MATH 371 - An Optimal Diet

Justin Overstreet, Austin Spurlin, and Jonas Smith

Spring 2020

We are assuming the subject is Male, 25-30, and consumes approximately two thousand calories a day. For some reason this person can only eat his meals at sonic during this given day. He would also really like to maximize the amount of fiber in his diet as he eats far too much fast food, and is feeling the effects. Based on the USDA[2] daily requirements for the Calories, Calories of fat, Fat, Saturated Fat, Trans Fat, Cholesterol, Sodium[1], Carbohydrates (Carbs), Sugar, and Protein, we decided on these constraints.

Calories	$\geq$	2000
Calories from Fat	$\geq$	585
Fat	$\geq$	64.5
Saturated Fat	$\leq$	27
Trans Fat	$\leq$	27
Cholesterol	$\leq$	250
Sodium	$\leq$	3000
Carbs	$\geq$	130
Sugar	$\leq$	50
Protein	$\geq$	56
Max. item count	$=$	4

Using Sonic Nutrition Brochure[1] we loaded the entirety of the Sonic menu into an excel sheet. And will be using all three hundred and sixty four menu items to decide what will maximize our fiber intake while only ordering four items.

[1]we adjusted the sodium[3] slightly (500mg) to increase the likely hood we would get a feasible solution.

Solution

We will be maximizing the following equation,

$$z = 3x_1 + 2x_2 + 2x_3 + 3x_4 + 3x_5 + ... + +2x_{363} + 4x_{364}$$

subject to,

$$\begin{cases} c_1x_1 + c_2x_2 + c_3x_3 + ... + c_{364}x_{364} \geq 2000 & \text{, where } c_i \text{ is the given Calories from the } x_{ith} \text{ item} \\ c_1x_1 + c_2x_2 + c_3x_3 + ... + c_{364}x_{364} \geq 585 & \text{, where } c_i \text{ is the given Calories from Fat from the } x_{ith} \text{ item} \\ c_1x_1 + c_2x_2 + c_3x_3 + ... + c_{364}x_{364} \geq 64.5 & \text{, where } c_i \text{ is the given Fat from the } x_{ith} \text{ item} \\ c_1x_1 + c_2x_2 + c_3x_3 + ... + c_{364}x_{364} \leq 27 & \text{, where } c_i \text{ is the given Saturate Fat from the } x_{ith} \text{ item} \\ c_1x_1 + c_2x_2 + c_3x_3 + ... + c_{364}x_{364} \leq 27 & \text{, where } c_i \text{ is the given Trans Fat from the } x_{ith} \text{ item} \\ c_1x_1 + c_2x_2 + c_3x_3 + ... + c_{364}x_{364} \leq 250 & \text{, where } c_i \text{ is the given Cholesterol from the } x_{ith} \text{ item} \\ c_1x_1 + c_2x_2 + c_3x_3 + ... + c_{364}x_{364} \leq 3000 & \text{, where } c_i \text{ is the given Sodium from the } x_{ith} \text{ item} \\ c_1x_1 + c_2x_2 + c_3x_3 + ... + c_{364}x_{364} \geq 130 & \text{, where } c_i \text{ is the given Carbs from the } x_{ith} \text{ item} \\ c_1x_1 + c_2x_2 + c_3x_3 + ... + c_{364}x_{364} \leq 50, & \text{, where } c_i \text{ is the given Sugar from the } x_{ith} \text{ item} \\ c_1x_1 + c_2x_2 + c_3x_3 + ... + c_{364}x_{364} \geq 56 & \text{, where } c_i \text{ is the given Protieen from the } x_{ith} \text{ item} \\ c_1x_1 + c_2x_2 + c_3x_3 + ... + c_{364}x_{364} = 4 & \text{, where } c_i \text{ is the given Item Count from the } x_{ith} \text{ item} \end{cases}$$

We used python, and the library PuLP[5] (a Linear Programming modeler written in python) to solve the LPP we created for our Sonic diet. We then imported the information from the Sonic Nutrition Brochure, into an excel sheet and loaded them into dictionaries (Show below).

```
df = pd.read_excel("nutrition_info.xlsx", nrows=365)
...
food_items = list(df['Food Items'])
...
column_names = df.head()
calories = dict(zip(food_items, df['Calories']))
calories_from_fat = dict(zip(food_items, df['Calories from Fat']))
fat = dict(zip(food_items, df['Fat (g)']))
saturated_fat = dict(zip(food_items, df['Saturated Fat (g)']))
trans_fat = dict(zip(food_items, df['Trans Fat (g)']))
cholesterol = dict(zip(food_items, df['Cholesterol (mg)']))
sodium = dict(zip(food_items, df['Sodium (mg)']))
carbs = dict(zip(food_items, df['Carbs (g)']))
dietary_fiber = dict(zip(food_items, df['Dietary Fiber (g)']))
sugar = dict(zip(food_items, df['Sugar (g)']))
protein = dict(zip(food_items, df['Protein (g)']))
item_count = dict(zip(food_items, df['Item Count']))
```

We then use PuLP to solve these equations.

```
# Define the scope of the optimization solution, i.e. the variables will be nonnegative
# in this case.
food_vars = LpVariable.dicts("Food", food_items, lowBound=0, cat=LpInteger)

# The objective function: dietary_fiber was selected for maximization.
prob += lpSum([dietary_fiber[i]*food_vars[i] for i in food_items])

# Calories
prob += lpSum([calories[f] * food_vars[f] for f in food_items]) >= 2000,
    "CalorieMinimum"
            .
            .
            .
# loading the other 8 constraints
            .
            .
            .
# Item Count
prob += lpSum([item_count[f] * food_vars[f] for f in food_items]) == 4,
    "ItemCountEquals"

prob.solve()
```

After PuLP is finished solving we are given the solution.

Status: Optimal

CRISPY TENDERS 5 PC. = 1
FRIES LARGE = 2
HANDMADE ONION RINGS LARGE =1

Maximized dietary fiber amount: 20.0 (g)

Other Nutrient Values:
Calories: 2170.0 kCal
Calories from Fat: 930.0 kCal
Fat: 103.0 (g)
Saturated Fat: 17.0 (g)
Trans Fat: 0.5 (g)
Cholesterol: 95.0 (mg)
Sodium: 2980.0 (mg)
Carbs: 254.0 (g)
Sugar: 28.0 (g)
Protein: 56.0 (g)

References

[1] Sonic Nurtition Brochure,
`https://offers.sonicdrivein.com/nutrition/nutrition.pdf`

[2] Macronutrients, ie {Carbs, Protein, Fats, Cholesteral, Fiber.. }
`https://www.nal.usda.gov/fnic/macronutrients`

[3] Caloric Intake,
`https://www.webmd.com/diet/features/estimated-calorie-requirement`

[4] Fat Intake,
`https://healthfully.com/262833-recommended-daily-intake-of-fat-calories.html`

[5] PuLP, a python linear programming library
`https://pythonhosted.org/PuLP/`