ISYE-6501
Fall 2024

Homework 11

Question 15.2

In the videos, we saw the "diet problem". In this homework you get to solve a diet problem with real data. The data is given in the file diet.xls.

1. **Formulate an optimization model (a linear program) to find the cheapest diet that satisfies the maximum and minimum daily nutrition constraints, and solve it using PuLP. Turn in your code and the solution. (The optimal solution should be a diet of air-popped popcorn, poached eggs, ranges, raw iceberg lettuce, raw celery, and frozen broccoli. UGH!)**

Analysis:

After running the optimization problem, I was able to find the same solution as listed in the question prompt.

"

52.64371 units of amount_Celery,_Raw
0.25960653 units of amount_Frozen_Broccoli
63.988506 units of amount_Lettuce,Iceberg,Raw
2.2929389 units of amount_Oranges
0.14184397 units of amount_Poached_Eggs
13.869322 units of amount_Popcorn,Air_Popped
Total cost of food = $4.34

"

Code:

```
from pulp import *
import pandas as pd

data = pd.read_excel(r"C:\Users\tmesaros\Desktop\Personal\data
15.2\diet.xls", sheet_name="Sheet1")
print(data.tail())
data = data[0:64]

data = data.values.tolist()

#Exercise 15.2 Question 1.

#Setting max and min intakes
min_in = [1500, 30, 20, 800, 130, 125, 60, 1000, 400, 700, 10]
max_in = [2500, 240, 70, 2000, 450, 250, 100, 10000, 5000, 1500, 40]

#Creating dictionary of foods
```

```
foods = [x[0] for x in data]
food_dict = []
for i in range(3, 14):
    food_dict.append(dict([(x[0], float(x[i])) for x in data]))

#Define the cost dictionary.
cost = dict([(x[0], float(x[1])) for x in data])

#Define optimization problem
problem = LpProblem('myProblem', LpMinimize)

#Create the optimization problem variables
var_food = LpVariable.dicts("foods", foods, 0)
var_chosen = LpVariable.dicts("chosen", foods, 0, 1, LpBinary)
amount = LpVariable.dicts("amount", foods, 0)

#Create the objective function
problem += lpSum([cost[food] * amount[food] for food in foods])

#Add nutrient constraints
for i in range(0, 10):
    nutrients = pulp.lpSum([food_dict[i][food] * amount[food] for food in
foods])
    problem += max_in[i] >= nutrients
    problem += min_in[i] <= nutrients

#Solve the optimization problem
problem.solve()
print('Solution:')

#Iterate through the solution variables and find if the food is in the
solution
for var in problem.variables():
    if var.varValue > 0:
        if str(var).find('chosen'):
            print(str(var.varValue) + " units of " + str(var))
print("Total cost of food = $%.2f" % value(problem.objective)
```

2. **Please add to your code the following constraints (which might require adding more variables) and solve the new model:**
    a. **If a food is selected, then a minimum of 1/10 serving must be chosen.**
    b. **Many people dislike celery and frozen broccoli. So at most one, but not both, can be selected.**
    c. **To get day-to-day variety in protein, at least 3 kinds of meat/poultry/fish/eggs must be selected.**

ANALYSIS:

After running the code below with the constraints listed in 2a, b, and c. I was able to get the following diet. "52.64371 Units of amount_Celery,_Raw
0.25960653 Units of amount_Frozen_Broccoli
63.988506 Units of amount_Lettuce,Iceberg,Raw
2.2929389 Units of amount_Oranges
0.14184397 Units of amount_Poached_Eggs
13.869322 Units of amount_Popcorn,Air_Popped
0.1 Units of foods_Bologna,Turkey
0.1 Units of foods_Frankfurter,_Beef
0.1 Units of foods_Ham,Sliced,Extralean
0.1 Units of foods_Hamburger_W_Toppings
0.1 Units of foods_Hotdog,_Plain
0.1 Units of foods_Kielbasa,Prk
0.1 Units of foods_Poached_Eggs
0.1 Units of foods_Pork
0.1 Units of foods_Roasted_Chicken
0.1 Units of foods_Scrambled_Eggs
0.1 Units of foods_White_Tuna_in_Water
Total cost of food = $4.34
"

I do not think it is correct because of how close it is to the original solution.

CODE:

```
#Create optimization problem for question 2abc
problem2abc = LpProblem('MyProblem2a', LpMinimize)

#Create the objective function
problem2abc += lpSum([cost[food] * amount[food] for food in foods])

#Add nutrient constraints
```

```
for i in range(0, 10):
    nutrients = pulp.lpSum([food_dict[i][food] * amount[food] for food in
foods])
    problem2abc += max_in[i] >= nutrients
    problem2abc += min_in[i] <= nutrients

#Add mimumum 0.1 units of food constraint from a
for food in foods:
    problem2abc += var_food[food] >= 0.1 * var_chosen[food]

#Add constraint for broccoli and celery from b
problem2abc += var_chosen['Frozen Broccoli'] + var_chosen['Celery, Raw']
<= 1

#Add constraint for at least three types of food from c
problem2abc += var_chosen['Roasted Chicken'] + var_chosen['Poached Eggs']
+ \
  var_chosen['Scrambled Eggs'] + var_chosen['Frankfurter, Beef'] + \
  var_chosen['Kielbasa,Prk'] + var_chosen['Hamburger W/Toppings'] + \
  var_chosen['Hotdog, Plain'] + var_chosen['Pork'] + \
  var_chosen['Bologna,Turkey'] + var_chosen['Ham,Sliced,Extralean'] + \
  var_chosen['White Tuna in Water'] \
  >= 3

problem2abc.solve()
print("Solution for 2abc:")
for var in problem2abc.variables():
    if var.varValue > 0:
        if str(var).find('chosen'):
            print(str(var.varValue)+ " Units of " + str(var))
print("Total cost of food = $%.2f" % value(problem2abc.objective))
```