ISYE 6501 : Homework 11
4/7/2021

Table of Contents

15.2.1 Diet Problem for the cheapest cost

Using PulP, I created an optimization to determine the least amount of cost with the maximum and minimum daily nutrition constraints.

CODE with OUTPUT:

```
from pulp import *
import pandas as pd
# load diet data

#Ingest data
data = pd.read_excel(open('/Users/dianeroberts/Documents/ISYE6501 Intro to
Analytics Modeling/FA_SP_hw11/diet.xls', 'rb'), sheet_name='Sheet1')
data.head()

# clean data - remove last few summary rows
data = data[0:64]
# convert to list
d_list = data.values.tolist()
```

In [14]:

```
# intake dictionary
foods = [i[0] for i in d_list]
calories = dict([(i[0], float(i[3])) for i in d_list])
cholesterol = dict([(i[0], float(i[4])) for i in d_list])
totalFat = dict([(i[0], float(i[5])) for i in d_list])
sodium = dict([(i[0], float(i[6])) for i in d_list])
carbs = dict([(i[0], float(i[7])) for i in d_list])
fiber = dict([(i[0], float(i[8])) for i in d_list])
protien = dict([(i[0], float(i[9])) for i in d_list])
vitaminA = dict([(i[0], float(i[10])) for i in d_list])
vitaminC = dict([(i[0], float(i[11])) for i in d_list])
calcium = dict([(i[0], float(i[12])) for i in d_list])
iron = dict([(i[0], float(i[13])) for i in d_list])
```

In [15]:

```
# min and max intakes for all foods
min_intake = [1500, 30, 20, 800, 130, 125, 60, 1000, 400, 700, 10]
max_intake = [2500, 240, 70, 2000, 450, 250, 100, 10000, 5000, 1500, 40]
```

In [16]:

```
# contraints for each column
constraints = []
for j in range(0,11):
 constraints.append(dict([(i[0], float(i[j+3])) for i in d_list]))
```

In [17]:

```
# cost dictionary
cost = dict([(i[0], float(i[1])) for i in d_list])

# create the optimization problem framework - minimization problem
FoodProb = LpProblem('PulpLpMinimize', LpMinimize)
```

In [18]:

```
# define the continous variables
Food_vars = LpVariable.dicts("foods", foods,0)

# define the binary variables
Chosen_data = LpVariable.dicts("Chosen",foods,0,1,"Binary")

# dictionary of variables
i = LpVariable.dicts("i", foods, 0)
```

In [19]:

```
# objective function
FoodProb += lpSum([cost[f] * Food_vars[f] for f in foods])

# add constraints for all foods
for i in range(0,11):
 constraints_i = pulp.lpSum([constraints[i][j] * Food_vars[j] for j in
foods])
 condition1 = min_intake[i] <= + constraints_i
 FoodProb += condition1
```

In [20]:

```
for i in range(0,11):
 constraints_i = pulp.lpSum([constraints[i][j] * Food_vars[j] for j in
foods])
 condition2 = max_intake[i] >= + constraints_i
 FoodProb += condition2

# solve the optimization problem!
FoodProb.solve()

# print the foods of the optimal diet
print('Optimization Solution:')
Optimization Solution:
```

In [23]:

```
for var in FoodProb.variables():
 if var.varValue > 0:
     if str(var).find('Chosen'):
         print(str(var.varValue) + " units of " + str(var))

# print the costs of the optimal diet
print("Total cost = $%.2f" % value(FoodProb.objective))
52.64371 units of foods_Celery,_Raw
0.25960653 units of foods_Frozen_Broccoli
```

```
63.988506 units of foods_Lettuce,Iceberg,Raw
2.2929389 units of foods_Oranges
0.14184397 units of foods_Poached_Eggs
13.869322 units of foods_Popcorn,Air_Popped
Total cost = $4.34
```

15.2.1 Diet Problem for the cheapest cost

CODE with OUTPUT:

```
# Insert a constraint to eat at least 1/10 of each food
for f in foods:
    FoodProb += Food_vars[f] >= .1 * Chosen_data[f]

# Either Broccoli or Celery
FoodProb += Chosen_data['Frozen Broccoli'] + \
Chosen_data['Celery, Raw'] <= 1, 'At most one Broccoli / Celery'
```

In [35]:

```
FoodProb += Chosen_data['Roasted Chicken'] + Chosen_data['Poached Eggs'] + \
 Chosen_data['Scrambled Eggs'] + Chosen_data['Frankfurter, Beef'] + \
 Chosen_data['Kielbasa,Prk'] + Chosen_data['Hamburger W/Toppings'] + \
 Chosen_data['Hotdog, Plain'] + Chosen_data['Pork'] + \
 Chosen_data['Bologna,Turkey'] + Chosen_data['Ham,Sliced,Extralean'] + \
 Chosen_data['White Tuna in Water'] \
>=3
```

In [36]:

```
FoodProb.solve()

# print the foods of the optimal diet
print('Optimization Solution:')

for var in FoodProb.variables():
 if var.varValue > 0:
     if str(var).find('Chosen'):
         print(str(var.varValue) + " units of " + str(var))

# print the costs of the optimal diet
print("Total cost = $%.2f" % value(FoodProb.objective))
Optimization Solution:
0.1 units of foods_Bologna,Turkey
51.522935 units of foods_Celery,_Raw
0.22815709 units of foods_Frozen_Broccoli
66.955456 units of foods_Lettuce,Iceberg,Raw
2.3760495 units of foods_Oranges
0.1 units of foods_Poached_Eggs
13.847028 units of foods_Popcorn,Air_Popped
0.1 units of foods_Scrambled_Eggs
Total cost = $4.38
```