Question 15.2

In the videos, we saw the “diet problem”. (The diet problem is one of the first large-scale optimization problems to be studied in practice. Back in the 1930’s and 40’s, the Army wanted to meet the nutritional requirements of its soldiers while minimizing the cost.) 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, oranges, raw iceberg lettuce, raw celery, and frozen broccoli. UGH!)
2. Please add to your model 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. (Hint: now you will need two variables for each food *i*: whether it is chosen, and how much is part of the diet. You’ll also need to write a constraint to link them.)
   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. [If something is ambiguous (e.g., should bean-and-bacon soup be considered meat?), just call it whatever you think is appropriate – I want you to learn how to write this type of constraint, but I don’t really care whether we agree on how to classify foods!]

If you want to see what a more full-sized problem would look like, try solving your models for the file `diet_large.xls`, which is a low-cholesterol diet model (rather than minimizing cost, the goal is to minimize cholesterol intake). I don’t know anyone who’d want to eat this diet – the optimal solution includes dried chrysanthemum garland, raw beluga whale flipper, freeze-dried parsley, etc. – which shows why it’s necessary to add additional constraints beyond the basic ones we saw in the video!

[**Note**: there are many optimal solutions, all with zero cholesterol, so you might get a different one. It probably won’t be much more appetizing than mine.]

Response:

15.2.1 – Mathematical representation of the problem

Data variables →

$n_{ij} = amount\ of\ nutrient\ j\ per\ unit\ of\ food\ i$
$min_j = minimum\ amount\ of\ nutrient\ j\ required$
$max_j = minimum\ amount\ of\ nutrient\ j\ required$
$c_i = cost\ per\ unit\ of\ food\ i$
$f_i = amount\ of\ food\ eaten\ i$

Objective and Constraints →

$$\min \sum c_i f_i$$

$where:$

$$f_i \geq 0$$
$$\sum n_{ij} f_i \geq min_j$$
$$\sum n_{ij} f_i \leq max_j$$

Based on above conditions and using pulp function from python, linear programming is used to get an optimal solution:

Output:

```
lp.solve()
print('Optimal solution:')
for var in lp.variables():
    if var.varValue > 0:
        if str(var).find('Selected_food'):
            print(str(var.varValue) + ' units of ' + str(var).replace('foods_',''))
print('Total cost incurred for food = $%.2f' % value(lp.objective))
```

```
Optimal solution:
52.64371 units of Celery,_Raw
0.25960653 units of Frozen_Broccoli
63.988506 units of Lettuce,Iceberg,Raw
2.2929389 units of Oranges
0.14184397 units of Poached_Eggs
13.869322 units of Popcorn,Air_Popped
Total cost incurred for food = $4.34
```

As our objective function is minimizing the cost of the total intake of food, so we end up having more vegetables i.e., celery, broccoli, lettuce, oranges, poached eggs and popcorn. Minimum total cost incurred for food is $4.34.

15.2.2 – Mathematical representation additional constraints

Data variables →

$$b_i = binary\ digit\ if\ food\ i\ is\ eaten$$

Additional Constraints →

$$where:$$

$$f_i \geq 0.1 * b_i$$
$$b_i \geq 0.000000001 * f_i$$
$$y_{frozen,broccoli} + y_{celery,raw} \leq 1$$
$$\begin{aligned} y_{roasted\ chicken} &+ y_{poached\ eggs} + y_{scrambled\ eggs} + y_{bologna,turkey} \\ &+ y_{frankfurter,beef} + y_{ham,sliced,extralean} \\ &+ y_{kielbasa,prk} + y_{pizza\ w/pepperoni} + y_{hotdog,plain} \\ &+ y_{hamburger\ w/toppings} + y_{sardines\ in\ oil} + y_{pork} \\ &+ y_{white\ tuna\ in\ water} + y_{chicknoodl\ soup} \\ &+ y_{splt\ pea\&hamsoup} + y_{vegetbeef\ soup} \\ &+ y_{neweng\ clamchwd} + y_{new\ e\ clamchwd,w/mlk} \\ &+ y_{beanbacn\ soup,w/water} \geq 3 \end{aligned}$$

Based on above additional conditions and using pulp function from python, linear programming is used to get an optimal solution:

Output:

```
lp.solve()
print('Optimal solution:')
for var in lp.variables():
    if var.varValue > 0:
        if str(var).find('Selected_food'):
            print(str(var.varValue) + ' units of ' + str(var).replace('foods_',''))
print('Total cost incurred for food = $%.2f' % value(lp.objective))
```

```
Optimal solution:
0.1 units of Bologna,Turkey
51.695579 units of Celery,_Raw
72.437846 units of Lettuce,Iceberg,Raw
2.8305897 units of Oranges
0.1 units of Poached_Eggs
13.559747 units of Popcorn,Air_Popped
0.1 units of Scrambled_Eggs
Total cost incurred for food = $4.52
```

As our objective function is minimizing the cost of the total intake of food and with above additional constraints i.e, at least three meat kinds and minimum 0.1 units of food eaten, so we end up having at least 3 meat substitutes (Bologna turkey, poached egg and scrambled eggs) with minimum 0.1 food. Celery was included and broccoli was excluded from the diet. Minimum total cost incurred for food is $4.52.