Assignment 1 Code

January 14, 2024

[2]:

```
#load libraries
from pulp import LpVariable, LpProblem, LpStatus, value, LpMinimize

#Set up notebook to display multiple outputs in one cell
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
```

[3]:

```
# original LP model, 0 servings permitted

x1 = LpVariable("x1", 0, None) # x1>=0 #plant-based kodiak cakes
x2 = LpVariable("x2", 0, None) # x2>=0 #nut and cheese snacks
x3 = LpVariable("x3", 0, None) # x3>=0 #tortilla espanola
x4 = LpVariable("x4", 0, None) # x4>=0 #BOOST drink mix
x5 = LpVariable("x5", 0, None) # x5>=0 #madras lentils

prob = LpProblem("problem", LpMinimize)

# constraints (multiplied by 7 for weekly requirements)
prob += 440*x1 + 160*x2 + 320*x3 + 170*x4 + 510*x5 <= (5000*7) #sodium (mg)
prob += 230*x1 + 180*x2 + 150*x3 + 220*x4 + 150*x5 >= (2000*7) #energy (kcal)
prob += 14*x1 + 8*x2 + 5*x3 + 10*x4 + 7*x5 >= (50*7) #protein (g)
prob += 1*x1 + 0.1*x2 + 0.2*x3 + 6*x4 + 0.1*x5 >= (20*7) #vitamin D (mcg)
prob += 24*x1 + 150*x2 + 30*x3 + 260*x4 + 40*x5 >= (1300*7) #calcium (mg)
prob += 3*x1 + 0.4*x2 + 0.8*x3 + 3.6*x4 + 2.1*x5 >= (18*7) #iron (mg)
prob += 155*x1 + 80*x2 + 320*x3 + 290*x4 + 360*x5 >= (4700*7) #potassium (mg)

# objective function to minimize
prob += (6.29/8)*x1 + (13.59/16)*x2 + (5.49/6)*x3 + (33.99/24)*x4 + (2.49/2)*x5␣
 ↪#cost per serving

# solve the problem
status = prob.solve()
LpStatus[status]

# print the results
print(f"""Minimum occurs at the servings:
      {value(x1):.2f} of Kodiak Cakes,
      {value(x2):.2f} of Fruit & Nut Packs,
      {value(x3):.2f} of Tortilla Espanola,
      {value(x4):.2f} of BOOST Drink Mix,
      {value(x5):.2f} of Madras Lentils""")
print(f"""Minimum weekly cost per food:
      ${(6.29/8) * value(x1):.2f} of Kodiak Cakes,
      ${(13.59/16) * value(x2):.2f} of Fruit & Nut Packs,
      ${(5.49/6) * value(x3):.2f} of Tortilla Espanola,
      ${(33.99/24) * value(x4):.2f} of BOOST Drink Mix,
      ${(2.49/2) * value(x5):.2f} of Madras Lentils,
      ${((6.29/8) * value(x1) + (13.59/16) * value(x2) +
       (5.49/6) * value(x3) + (33.99/24) * value(x4) +
       (2.49/2) * value(x5)):.2f} in total""")
```

```
Welcome to the CBC MILP Solver
Version: 2.10.3
Build Date: Dec 15 2019

command line - /Users/baileyscoville/anaconda3/lib/python3.10/site-
packages/pulp/solverdir/cbc/osx/64/cbc /var/folders/z0/v3y1p30945d16_whz3lt8v_h0
000gn/T/e3e1911bdada4397a3f3270ed9954465-pulp.mps timeMode elapsed branch
printingOptions all solution /var/folders/z0/v3y1p30945d16_whz3lt8v_h0000gn/T/e3
e1911bdada4397a3f3270ed9954465-pulp.sol (default strategy 1)
At line 2 NAME          MODEL
At line 3 ROWS
At line 12 COLUMNS
At line 53 RHS
At line 61 BOUNDS
At line 62 ENDATA
Problem MODEL has 7 rows, 5 columns and 35 elements
Coin0008I MODEL read with 0 errors
Option for timeMode changed from cpu to elapsed
Presolve 7 (0) rows, 5 (0) columns and 35 (0) elements
0  Obj 0 Primal inf 582.00411 (6)
3  Obj 109.24169
Optimal - objective value 109.24169
Optimal objective 109.2416946 - 3 iterations time 0.002
Option for printingOptions changed from normal to all
Total time (CPU seconds):       0.00   (Wallclock seconds):       0.02
```

```
[3]: 'Optimal'
```

```
Minimum occurs at the servings:
      0.00 of Kodiak Cakes,
      0.00 of Fruit & Nut Packs,
      79.40 of Tortilla Espanola,
      25.84 of BOOST Drink Mix,
      0.00 of Madras Lentils
Minimum weekly cost per food:
      $0.00 of Kodiak Cakes,
      $0.00 of Fruit & Nut Packs,
      $72.65 of Tortilla Espanola,
      $36.59 of BOOST Drink Mix,
      $0.00 of Madras Lentils,
      $109.24 in total
```

[4]:

```
#measure original model results against weekly constraints
print(f"{(value(x3)*320) + (value(x4)*170):.0f} in sodium for weekly maximum of␣
  ↪{5000*7} mg")
print(f"{(value(x3)*150) + (value(x4)*220):.0f} in energy for weekly minimum of␣
  ↪{2000*7} kcal")
print(f"{(value(x3)*5) + (value(x4)*10):.0f} in protein for weekly minimum of␣
  ↪{50*7} g")
print(f"{(value(x3)*0.2) + (value(x4)*6):.0f} in vitamin D for weekly minimum␣
  ↪of {20*7} mcg")
print(f"{(value(x3)*30) + (value(x4)*260):.0f} in calcium for weekly minimum of␣
  ↪{1300*7} mg")
print(f"{(value(x3)*0.8) + (value(x4)*3.6):.0f} in iron for weekly minimum of␣
  ↪{18*7} mg")
print(f"{(value(x3)*320) + (value(x4)*290):.0f} in potassium for weekly minimum␣
  ↪of {4700*7} mg")
```

```
29799 in sodium for weekly maximum of 35000 mg
17594 in energy for weekly minimum of 14000 kcal
655 in protein for weekly minimum of 350 g
171 in vitamin D for weekly minimum of 140 mcg
9100 in calcium for weekly minimum of 9100 mg
157 in iron for weekly minimum of 126 mg
32900 in potassium for weekly minimum of 32900 mg
```

[5]:

```
# altered LP model, 1 serving of each food required

x1 = LpVariable("x1", 1, None) # x1>=1 #plant-based kodiak cakes
x2 = LpVariable("x2", 1, None) # x2>=1 #nut and cheese snacks
x3 = LpVariable("x3", 1, None) # x3>=1 #tortilla espanola
x4 = LpVariable("x4", 1, None) # x4>=1 #BOOST drink mix
x5 = LpVariable("x5", 1, None) # x5>=1 #madras lentils

prob = LpProblem("problem", LpMinimize)

# constraints (multiplied by 7 for weekly requirements)
prob += 440*x1 + 160*x2 + 320*x3 + 170*x4 + 510*x5 <= (5000*7) #sodium (mg)
prob += 230*x1 + 180*x2 + 150*x3 + 220*x4 + 150*x5 >= (2000*7) #energy (kcal)
prob += 14*x1 + 8*x2 + 5*x3 + 10*x4 + 7*x5 >= (50*7) #protein (g)
prob += 1*x1 + 0.1*x2 + 0.2*x3 + 6*x4 + 0.1*x5 >= (20*7) #vitamin D (mcg)
prob += 24*x1 + 150*x2 + 30*x3 + 260*x4 + 40*x5 >= (1300*7) #calcium (mg)
prob += 3*x1 + 0.4*x2 + 0.8*x3 + 3.6*x4 + 2.1*x5 >= (18*7) #iron (mg)
prob += 155*x1 + 80*x2 + 320*x3 + 290*x4 + 360*x5 >= (4700*7) #potassium (mg)

# objective function to minimize
prob += (6.29/8)*x1 + (13.59/16)*x2 + (5.49/6)*x3 + (33.99/24)*x4 + (2.49/2)*x5 #cost per serving

# solve the problem
status = prob.solve()
LpStatus[status]

# print the results
print(f"""Minimum occurs at the servings:
      {value(x1):.2f} of Kodiak Cakes,
      {value(x2):.2f} of Fruit & Nut Packs,
      {value(x3):.2f} of Tortilla Espanola,
      {value(x4):.2f} of BOOST Drink Mix,
      {value(x5):.2f} of Madras Lentils""")
print(f"""Minimum weekly cost per food:
      ${(6.29/8) * value(x1):.2f} of Kodiak Cakes,
      ${(13.59/16) * value(x2):.2f} of Fruit & Nut Packs,
      ${(5.49/6) * value(x3):.2f} of Tortilla Espanola,
      ${(33.99/24) * value(x4):.2f} of BOOST Drink Mix,
      ${(2.49/2) * value(x5):.2f} of Madras Lentils,
      ${((6.29/8) * value(x1) + (13.59/16) * value(x2) +
       (5.49/6) * value(x3) + (33.99/24) * value(x4) +
       (2.49/2) * value(x5)):.2f} in total""")
```

```
Welcome to the CBC MILP Solver
Version: 2.10.3
Build Date: Dec 15 2019

command line - /Users/baileyscoville/anaconda3/lib/python3.10/site-
packages/pulp/solverdir/cbc/osx/64/cbc /var/folders/z0/v3y1p30945d16_whz3lt8v_h0
000gn/T/48fa0f5b73f14e9381d6f7eb0c633f86-pulp.mps timeMode elapsed branch
printingOptions all solution /var/folders/z0/v3y1p30945d16_whz3lt8v_h0000gn/T/48
fa0f5b73f14e9381d6f7eb0c633f86-pulp.sol (default strategy 1)
At line 2 NAME          MODEL
At line 3 ROWS
At line 12 COLUMNS
At line 53 RHS
At line 61 BOUNDS
At line 67 ENDATA
Problem MODEL has 7 rows, 5 columns and 35 elements
Coin0008I MODEL read with 0 errors
Option for timeMode changed from cpu to elapsed
```

```
Presolve 7 (0) rows, 5 (0) columns and 35 (0) elements
0  Obj 5.211875 Primal inf 547.01769 (6)
3  Obj 110.02205
Optimal - objective value 110.02205
Optimal objective 110.0220466 - 3 iterations time 0.002
Option for printingOptions changed from normal to all
Total time (CPU seconds):       0.00   (Wallclock seconds):       0.00
```

[5]:

```
'Optimal'
```

```
Minimum occurs at the servings:
      1.00 of Kodiak Cakes,
      1.00 of Fruit & Nut Packs,
      78.15 of Tortilla Espanola,
      25.16 of BOOST Drink Mix,
      1.00 of Madras Lentils
Minimum weekly cost per food:
      $0.79 of Kodiak Cakes,
      $0.85 of Fruit & Nut Packs,
      $71.51 of Tortilla Espanola,
      $35.63 of BOOST Drink Mix,
      $1.25 of Madras Lentils,
      $110.02 in total
```

[6]:

```
#measure altered model results against weekly constraints
print(f"{(value(x1)*440) + (value(x2)*160) + (value(x3)*320) + (value(x4)*170)␣
 ↪+ (value(x5)*510):.0f} in sodium for weekly maximum of {5000*7} mg")
print(f"{(value(x1)*230) + (value(x2)*180) + (value(x3)*150) + (value(x4)*220)␣
 ↪+ (value(x5)*150):.0f} in energy for weekly minimum of {2000*7} kcal")
print(f"{(value(x1)*14) + (value(x2)*8) + (value(x3)*5) + (value(x4)*10) +␣
 ↪(value(x5)*7):.0f} in protein for weekly minimum of {50*7} g")
print(f"{(value(x1)*1) + (value(x2)*0.1) + (value(x3)*0.2) + (value(x4)*6) +␣
 ↪(value(x5)*0.1):.0f} in vitamin D for weekly minimum of {20*7} mcg")
print(f"{(value(x1)*24) + (value(x2)*150) + (value(x3)*30) + (value(x4)*260) +␣
 ↪(value(x5)*40):.0f} in calcium for weekly minimum of {1300*7} mg")
print(f"{(value(x1)*3) + (value(x2)*0.4) + (value(x3)*0.8) + (value(x4)*3.6) +␣
 ↪(value(x5)*2.1):.0f} in iron for weekly minimum of {18*7} mg")
print(f"{(value(x1)*155) + (value(x2)*80) + (value(x3)*320) + (value(x4)*290) +␣
 ↪(value(x5)*360):.0f} in potassium for weekly minimum of {4700*7} mg")
```

```
30396 in sodium for weekly maximum of 35000 mg
17818 in energy for weekly minimum of 14000 kcal
671 in protein for weekly minimum of 350 g
168 in vitamin D for weekly minimum of 140 mcg
9100 in calcium for weekly minimum of 9100 mg
159 in iron for weekly minimum of 126 mg
32900 in potassium for weekly minimum of 32900 mg
```

[ ]:

```
```