```
from pulp import *
import pandas as pd
import os

os.chdir('/Users/Hatim/Desktop/GATech/GATech-Introduction-to-Analytical-M

data = pd.read_excel('diet.xls')
data = data[0:64].values.tolist()


foods = [x[0] for x in data]
cost = dict([(x[0], float(x[1])) for x in data])
calories = dict([(x[0], float(x[3])) for x in data])
chol = dict([(x[0], float(x[4])) for x in data])
fat = dict([(x[0], float(x[5])) for x in data])
sodium = dict([(x[0], float(x[6])) for x in data])
carbs = dict([(x[0], float(x[7])) for x in data])
fiber = dict([(x[0], float(x[8])) for x in data])
protein = dict([(x[0], float(x[9])) for x in data])
vitA = dict([(x[0], float(x[10])) for x in data])
vitC = dict([(x[0], float(x[11])) for x in data])
calcium = dict([(x[0], float(x[12])) for x in data])
iron = dict([(x[0], float(x[13])) for x in data])

diet = LpProblem("Diet Optimization",LpMinimize)

#set the initial variables
foodVars = LpVariable.dicts("Foods", foods, lowBound = 0 )
chosenVars = LpVariable.dicts("Chosen", foods, lowBound = 0, upBound = 1,

#Add the objective function to mimimize the total cost
diet += lpSum([cost[f]*foodVars[f] for f in foods]), "Total Cost"

#Add in the constraints
diet += lpSum([calories[f]*foodVars[f] for f in foods]) >= 1500, 'min Cal
diet += lpSum([calories[f]*foodVars[f] for f in foods]) <= 2500, 'max Cal

diet += lpSum([chol[f]*foodVars[f] for f in foods]) >= 30, 'min Cholester
diet += lpSum([chol[f]*foodVars[f] for f in foods]) <= 240, 'max Choleste

diet += lpSum([fat[f]*foodVars[f] for f in foods]) >= 20, 'min fat'
diet += lpSum([fat[f]*foodVars[f] for f in foods]) <= 70, 'max fat'

diet += lpSum([sodium[f]*foodVars[f] for f in foods]) >= 800, 'min sodium
diet += lpSum([sodium[f]*foodVars[f] for f in foods]) <= 2000, 'max sodiu
```

```
diet += lpSum([carbs[f]*foodVars[f] for f in foods]) >= 130, 'min Carbs'
diet += lpSum([carbs[f]*foodVars[f] for f in foods]) <= 450, 'max Carbs'

diet += lpSum([fiber[f]*foodVars[f] for f in foods]) >= 125, 'min fiber'
diet += lpSum([fiber[f]*foodVars[f] for f in foods]) <= 250, 'max fiber'

diet += lpSum([protein[f]*foodVars[f] for f in foods]) >= 60, 'min protei
diet += lpSum([protein[f]*foodVars[f] for f in foods]) <= 100, 'max prote

diet += lpSum([vitA[f]*foodVars[f] for f in foods]) >= 1000, 'min vitA'
diet += lpSum([vitA[f]*foodVars[f] for f in foods]) <= 10000, 'max vitA'

diet += lpSum([vitC[f]*foodVars[f] for f in foods]) >= 400, 'min vitC'
diet += lpSum([vitC[f]*foodVars[f] for f in foods]) <= 5000, 'max vitC'

diet += lpSum([calcium[f]*foodVars[f] for f in foods]) >= 700, 'min calci
diet += lpSum([calcium[f]*foodVars[f] for f in foods]) <= 1500, 'max calc

diet += lpSum([iron[f]*foodVars[f] for f in foods]) >= 10, 'min iron'
diet += lpSum([iron[f]*foodVars[f] for f in foods]) <= 40, 'max iron'

diet.solve()
print("Status:", LpStatus[diet.status])
for v in diet.variables():
    if v.varValue != 0.0:
        print(v.name, "=", v.varValue)



for f in foods:
     diet += foodVars[f] <= 10000000*chosenVars[f]
     diet += foodVars[f] >= .1*chosenVars[f]

diet += chosenVars['Frozen Broccoli'] + chosenVars['Celery, Raw'] <=1

diet += chosenVars['Tofu'] + chosenVars['Roasted Chicken'] + \
chosenVars['Poached Eggs']+chosenVars['Scrambled Eggs']+chosenVars['Bolog
+chosenVars['Frankfurter, Beef']+chosenVars['Ham,Sliced,Extralean'] \
+chosenVars['Kielbasa,Prk']+chosenVars['Hamburger W/Toppings'] \
+chosenVars['Hotdog, Plain']+chosenVars['Pork'] +chosenVars['Sardines in
+chosenVars['White Tuna in Water'] >= 3
```

```
diet.solve()
print("Status:", LpStatus[diet.status])
for v in diet.variables():
    if v.varValue != 0.0: #Only print items that are not zero
        print(v.name, "=", v.varValue)

print ("Total Cost of food with additiona constraints is $%.2f" % value(d
```