15.2.1

March 28, 2018

```
In [1]: #import pulp and pandas
        from pulp import *
        import pandas as pd
        import xlrd
```

```
In [ ]: #test pulp
        pulp.pulpTestAll()
```

```
In [2]: #read in food data
        data = pd.read_excel('diet.xls',
                             skip_footer=3)
```

```
In [3]: # convert to list to make dictionaries later
        data = data.values.tolist()
```

```
In [4]: #read in requirements
        requirements = pd.read_excel('diet.xls',
                                     skiprows=list(range(1,66)),
                                     usecols=[2,3,4,5,6,7,8,9,10,11,12,13],
                                     header=0,
                                     index_col=0)
```

```
In [5]: requirements
```

```
Out[5]:                     Calories  Cholesterol mg  Total_Fat g  Sodium mg  \
        Serving Size
        Minimum daily intake     1500              30           20        800
        Maximum daily intake     2500             240           70       2000

                            Carbohydrates g  Dietary_Fiber g  Protein g  Vit_A IU  \
        Serving Size
        Minimum daily intake              130              125         60      1000
        Maximum daily intake              450              250        100     10000

                            Vit_C IU  Calcium mg  Iron mg
        Serving Size
        Minimum daily intake      400         700       10
        Maximum daily intake     5000        1500       40
```

```
In [6]: # Use .loc to get specific requirements when building model
        # Ex. 'Calories' and 'Minimum daily intake'
        requirements.loc['Minimum daily intake','Calories']
```

```
Out[6]: 1500
```

```
In [7]: requirements.loc['Maximum daily intake','Calories']
```

```
Out[7]: 2500
```

```
In [8]: #create list of foods, then dictionaries for each nutrient with food as key and amount

        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])
        cholesterol = 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])
```

```
In [9]: # Initialize Pulp Optimization Object
        diet = LpProblem('diet', LpMinimize)
```

```
In [10]: #Create food variables
         foodvars = LpVariable.dict('Foods', foods, 0)
```

```
In [11]: #Write Objective Function
         diet += lpSum([cost[f] * foodvars[f] for f in foods]), 'Total Cost'
```

```
In [12]: # add in additional constraints for max and min nutrients
         diet += lpSum([calories[f] * foodvars[f] for f in foods]) >= requirements.loc['Minimu
         diet += lpSum([calories[f] * foodvars[f] for f in foods]) <= requirements.loc['Maximu
```

```
In [13]: diet += lpSum([cholesterol[f] * foodvars[f] for f in foods]) >= requirements.loc['Mini
         diet += lpSum([cholesterol[f] * foodvars[f] for f in foods]) <= requirements.loc['Maxi
```

```
In [14]: diet += lpSum([fat[f] * foodvars[f] for f in foods]) >= requirements.loc['Minimum dail
         diet += lpSum([fat[f] * foodvars[f] for f in foods]) <= requirements.loc['Maximum dail
```

```
In [15]: diet += lpSum([sodium[f] * foodvars[f] for f in foods]) >= requirements.loc['Minimum d
         diet += lpSum([sodium[f] * foodvars[f] for f in foods]) <= requirements.loc['Maximum d
```

```
In [16]: diet += lpSum([carbs[f] * foodvars[f] for f in foods]) >= requirements.loc['Minimum da
         diet += lpSum([carbs[f] * foodvars[f] for f in foods]) <= requirements.loc['Maximum da
```

```
In [17]: diet += lpSum([fiber[f] * foodvars[f] for f in foods]) >= requirements.loc['Minimum da
         diet += lpSum([fiber[f] * foodvars[f] for f in foods]) <= requirements.loc['Maximum da
```

```
In [18]: diet += lpSum([protein[f] * foodvars[f] for f in foods]) >= requirements.loc['Minimum
         diet += lpSum([protein[f] * foodvars[f] for f in foods]) <= requirements.loc['Maximum
```

```
In [19]: diet += lpSum([vita[f] * foodvars[f] for f in foods]) >= requirements.loc['Minimum da
         diet += lpSum([vita[f] * foodvars[f] for f in foods]) <= requirements.loc['Maximum da
```

```
In [20]: diet += lpSum([vitc[f] * foodvars[f] for f in foods]) >= requirements.loc['Minimum da
         diet += lpSum([vitc[f] * foodvars[f] for f in foods]) <= requirements.loc['Maximum da
```

```
In [21]: diet += lpSum([calcium[f] * foodvars[f] for f in foods]) >= requirements.loc['Minimum
         diet += lpSum([calcium[f] * foodvars[f] for f in foods]) <= requirements.loc['Maximum
```

```
In [22]: diet += lpSum([iron[f] * foodvars[f] for f in foods]) >= requirements.loc['Minimum da
         diet += lpSum([iron[f] * foodvars[f] for f in foods]) <= requirements.loc['Maximum da
```

```
In [23]: #Solve and check status
         diet.solve()
         LpStatus[diet.status]
```

```
Out[23]: 'Optimal'
```

```
In [24]: #Print foods and their amounts for an optimal diet
         for v in diet.variables():
             if v.varValue>0:
                 print (v.name, "=", v.varValue)
```

```
Foods_Celery,_Raw = 52.64371
Foods_Frozen_Broccoli = 0.25960653
Foods_Lettuce,Iceberg,Raw = 63.988506
Foods_Oranges = 2.2929389
Foods_Poached_Eggs = 0.14184397
Foods_Popcorn,Air_Popped = 13.869322
```