Appendix

In [1]:

```
#user input data
age = int(input("Please enter your age(integer):"))
gender = input("Please enter your gender(male/female):")
gender = gender+'_'
act = input("Please choose your physical activity level(sedentary/moderately_active/act
ive):")
```

```
Please enter your age(integer):22
Please enter your gender(male/female):female
Please choose your physical activity level(sedentary/moderately_active/act
ive):sedentary
```

In [2]:

```
#read in user data
if 18 <= age <=25:
    age = '18-25_'
elif 26 <= age <= 30:
    age = '26-30_'
elif age > 30:
    age = 'over_30_'
user = age + gender + act
user
```

Out[2]:

```
'18-25_female_sedentary'
```

In [3]:

```
from gurobipy import *
import numpy as np
import pandas as pd
constraints = pd.read_excel('OR Project Data Constraints.xlsx', index_col = 0, header=
0)
n,m = constraints.shape
```

In [4]:

```
req= constraints[user]
```

In [5]:

```
#########Parameters Set-up############
# Read the data from the csv file and use the first column as the index of rows

project = pd.read_csv('data.csv')
N,M = project.shape
print(project.columns)
```

```
Index(['Category', 'Food Name', 'Energy (kcal)', 'Protein (g)',
       'Total fat (g)', 'Saturated fat (g)', 'Dietary fibre (g)',
       'Carbohydrate (g)', 'Cholesterol (mg)', 'Sodium (mg)', 'Sugar(g)'],
      dtype='object')
```

In [6]:

```
project.shape
```

Out[6]:

```
(136, 11)
```

In [7]:

```
nutrient = project.iloc[0:N, 2:M]
nutrient.head()
```

Out[7]:

	Energy (kcal)	Protein (g)	Total fat (g)	Saturated fat (g)	Dietary fibre (g)	Carbohydrate (g)	Cholesterol (mg)	Sodium (mg)	Sugar(g)
0	323.71	28.11	23.43	9.37	0.00	0.00	153.34	1316.15	0.00
1	318.04	25.14	7.00	1.85	1.74	36.57	76.03	1969.63	4.63
2	556.47	13.16	26.74	10.05	1.54	65.77	205.80	1937.08	13.62
3	695.50	27.30	39.65	14.24	3.90	57.85	104.00	7904.00	2.18
4	668.10	18.60	51.09	20.70	0.00	33.54	366.80	772.90	2.16

In [8]:

```
#for objective function
total_fat = nutrient.iloc[0:N,2]
saturated_fat = nutrient.iloc[0:N,3]
cholesterol = nutrient.iloc[0:N,6]
sodium = nutrient.iloc[0:N,7]
sugar = nutrient.iloc[0:N,8]
```

In [9]:

```
req
```

Out[9]:

```
Nutrient
Energy (Kcal)        1600.000000
Protein (g)            44.000000
Total fat              64.000000
Saturated fat          20.000000
Fiber (g)              23.000000
Carbohydrate (g)      120.000000
Cholesterol           250.000000
Sodium               2200.000000
sugar                  41.375743
Name: 18-25_female_sedentary, dtype: float64
```

In [10]:

```
m = Model("food")
D=5

# Creat variables
x = m.addVars(N, D, vtype=GRB.BINARY, name = "x")

# Set objective
#for penalty function
#m.setObjective(quicksum(x[i,d]*total_fat[i]/req[2]+x[i,d]*saturated_fat[i]/req[3]+x[i,
d]*cholesterol[i]/req[6]+x[i,d]*sodium[i]/req[7]+x[i,d]*sugar[i]/req[8] for i in range
(N) for d in range(D)), GRB.MINIMIZE)
#minimize total fat
#m.setObjective(quicksum(x[i,d]*total_fat[i] for i in range(N) for d in range(D)), GRB.
MINIMIZE)
#minimize saturated_fat
#m.setObjective(quicksum(x[i,d]*saturated_fat[i] for i in range(N) for d in range(D)),
 GRB.MINIMIZE)
#minimize cholesterol
#m.setObjective(quicksum(x[i,d]*cholesterol[i] for i in range(N) for d in range(D)), GR
B.MINIMIZE)
#minimize sodium
#m.setObjective(quicksum(x[i,d]*sodium[i] for i in range(N) for d in range(D)), GRB.MIN
IMIZE)
#minimize sugar
m.setObjective(quicksum(x[i,d]*sugar[i] for i in range(N) for d in range(D)), GRB.MINIM
IZE)

# Add constraints:
for d in range(D):
    m.addConstr( quicksum(x[i,d] for i in project[project.Category == 'Main'].index) ==
2, name = "mains")
    m.addConstr( quicksum(x[i,d] for i in project[project.Category == 'Breakfast'].inde
x) ==1, name = "breakfast")
    m.addConstr( quicksum(x[i,d] for i in  project[project.Category == 'Fruit'].index)
<=2, name = "fruits")
    m.addConstr( quicksum(x[i,d] for i in  project[project.Category == 'Dessert & Snac
k'].index) <=1, name = "dessert")
    m.addConstr( quicksum(x[i,d] for i in  project[project.Category == 'Beverage'].inde
x) <=2, name = "bev")

    m.addConstrs( (quicksum(x[i,d]*nutrient.iloc[i, j] for i in range(N)) >= req[j] for
j in [0,1,4,5])
             , name = "min_req")
    m.addConstrs( (quicksum(x[i,d]*nutrient.iloc[i, j] for i in range(N)) <= req[j] for
j in [2,3,6,7,8])
             , name = "max_intake")

for i in range(N):
    m.addConstr( quicksum(x[i,d] for d in range(D)) <=2, name = "max_repitition")
```

```
Academic license - for non-commercial use only
```

In [11]:

```
project.iloc[1,2]
```

Out[11]:

```
318.04
```

In [11]:

```
# Solving the model
m.setParam( 'OutputFlag', False )
%time m.optimize()

# #  Print optimal solutions and optimal value
# for v in m.getVars():
#     if(v.x == 1):
#         print(v.VarName, v.x)

# print('Obj:', m.objVal)
```

```
Wall time: 2min 11s
```

In [12]:

```
#  Print optimal x for x nonzero and optimal value
s_edge = []
for v in x:
    if x[v].x > 0.001:
        print(x[v].VarName, x[v].x)
        #add both of the indicies by 1
        edge = np.add(v, (1,1))
        #append the edge to the resulting list of edges
        s_edge.append(edge)

print('Obj:', m.objVal)
print(s_edge)
```

```
x[9,3] 1.0
x[14,1] 1.0
x[14,4] 1.0
x[21,3] 1.0
x[22,0] 1.0
x[22,2] 1.0
x[36,1] 1.0
x[36,4] 1.0
x[48,0] 1.0
x[48,2] 1.0
x[58,1] 1.0
x[59,3] 1.0
x[63,0] 1.0
x[63,2] 1.0
x[76,4] 1.0
x[86,3] 1.0
x[86,4] 1.0
x[88,3] 1.0
x[100,1] 1.0
x[100,3] 1.0
x[102,0] 1.0
x[102,4] 1.0
x[104,1] 1.0
x[104,4] 1.0
x[105,2] 1.0
x[105,3] 1.0
x[115,0] 1.0
x[115,3] 1.0
x[116,1] 1.0
x[116,4] 1.0
x[135,2] 1.0
Obj: 119.27
[array([10,  4]), array([15,  2]), array([15,  5]), array([22,  4]), array
([23,  1]), array([23,  3]), array([37,  2]), array([37,  5]), array([49,
1]), array([49,  3]), array([59,  2]), array([60,  4]), array([64,  1]), a
rray([64,  3]), array([77,  5]), array([87,  4]), array([87,  5]), array
([89,  4]), array([101,   2]), array([101,   4]), array([103,   1]), array
([103,   5]), array([105,   2]), array([105,   5]), array([106,   3]), arr
ay([106,   4]), array([116,   1]), array([116,   4]), array([117,   2]), a
rray([117,   5]), array([136,   3])]
```

In [18]:

```
project.iloc[[22,48,63,102,115]]
```

Out[18]:

	Category	Food Name	Energy (kcal)	Protein (g)	Total fat (g)	Saturated fat (g)	Dietary fibre (g)	Carbohydrate (g)	Choleste (n
22	Main	Claypot rice with salted fish, chicken and chi...	527.00	1.62	5.29	1.60	1.47	3.97	
48	Main	Beef and Teriyaki Chicken Bento	743.28	43.03	20.88	7.92	10.76	95.35	
63	Dessert & Snack	Beancurd Dessert with Red Bean Toppings	164.64	9.80	5.25	2.63	7.45	19.60	
102	Fruit	Passion fruit	17.65	0.40	0.13	0.01	1.89	4.26	
115	Breakfast	Plain porridge	207.48	6.55	1.64	0.00	2.18	42.04	

In [19]:

```
project.iloc[[14,36,58,100,104,116]]
```

Out[19]:

	Category	Food Name	Energy (kcal)	Protein (g)	Total fat (g)	Saturated fat (g)	Dietary fibre (g)	Carbohydrate (g)	Chol
14	Main	Nasi Lemak with chicken wing	656.58	26.08	24.94	10.65	6.52	81.21	
36	Main	Salmon and Teriyaki Chicken Bento	664.54	28.54	16.99	5.17	8.03	99.90	
58	Dessert & Snack	Pan-fried turnip cake	61.64	0.71	3.73	1.41	0.41	6.35	
100	Fruit	Dragonfruit	105.84	2.81	0.22	0.04	5.83	22.90	
104	Fruit	Mandarin orange	38.54	0.96	0.26	0.04	1.58	7.97	
116	Breakfast	Red rice porridge, plain	74.24	2.30	0.51	0.15	1.02	15.10	

In [15]:

```
project.iloc[[22,48,63,105,135]]
```

Out[15]:

	Category	Food Name	Energy (kcal)	Protein (g)	Total fat (g)	Saturated fat (g)	Dietary fibre (g)	Carbohydrate (g)	Choleste (n
22	Main	Claypot rice with salted fish, chicken and chi...	527.00	1.62	5.29	1.60	1.47	3.97	0
48	Main	Beef and Teriyaki Chicken Bento	743.28	43.03	20.88	7.92	10.76	95.35	0
63	Dessert & Snack	Beancurd Dessert with Red Bean Toppings	164.64	9.80	5.25	2.63	7.45	19.60	0
105	Fruit	Orange	46.48	1.11	0.11	0.00	2.66	8.88	0
135	Breakfast	Pow, char siew, steamed	160.59	5.78	6.47	2.44	0.74	19.77	10

In [16]:

```
project.iloc[[9,21,59,86,88,100,105,115]]
```

Out[16]:

	Category	Food Name	Energy (kcal)	Protein (g)	Total fat (g)	Saturated fat (g)	Dietary fibre (g)	Carbohydrate (g)	Cholest (
9	Main	Chicken rice with steamed chicken	557.70	28.05	13.86	4.95	3.30	80.19	3
21	Main	Chinese rojak	517.58	16.29	21.48	6.06	6.84	63.85	1
59	Dessert & Snack	Vietnam spring roll	78.95	4.53	1.80	0.69	0.43	11.14	
86	Beverage	Soya bean milk, without sugar	82.50	6.75	5.00	0.65	2.00	3.00	
88	Beverage	Brewed coffee	3.70	0.16	0.00	0.00	0.00	0.62	
100	Fruit	Dragonfruit	105.84	2.81	0.22	0.04	5.83	22.90	
105	Fruit	Orange	46.48	1.11	0.11	0.00	2.66	8.88	
115	Breakfast	Plain porridge	207.48	6.55	1.64	0.00	2.18	42.04	

In [17]:

```
project.iloc[[14, 36, 76, 86, 102, 104, 116]]
```

Out[17]:

	Category	Food Name	Energy (kcal)	Protein (g)	Total fat (g)	Saturated fat (g)	Dietary fibre (g)	Carbohydrate (g)	Choleste (m
14	Main	Nasi Lemak with chicken wing	656.58	26.08	24.94	10.65	6.52	81.21	117.
36	Main	Salmon and Teriyaki Chicken Bento	664.54	28.54	16.99	5.17	8.03	99.90	0.
76	Dessert & Snack	Yoghurt muesli bar	123.38	2.05	4.31	2.88	1.98	18.45	3.
86	Beverage	Soya bean milk, without sugar	82.50	6.75	5.00	0.65	2.00	3.00	0.
102	Fruit	Passion fruit	17.65	0.40	0.13	0.01	1.89	4.26	0.
104	Fruit	Mandarin orange	38.54	0.96	0.26	0.04	1.58	7.97	0.
116	Breakfast	Red rice porridge, plain	74.24	2.30	0.51	0.15	1.02	15.10	0.

In [ ]: