IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Solution And Sensitivity Analysis Of Diet Problem Of Indian Teenage Girls Using Python Programming

Pratibha Lakhani : Associate Professor, Department of Statistics, Government Vidarbha Institute of Science and Humanities, Amravati (MS), India.

Pooja Wadodkar : Assistant Professor, Kaveri college of Arts, Science and Commerce, Pune (MS), India.

Abstract:

The Linear Programming Problem (LPP) is a problem of finding the optimal value of the given linear function. The optimal value can be either maximum value or minimum value. Python (Spider) is a widely used programming language and freely available software for statistical computing. This paper is the study of determining the optimum solution for the diet problem of the teenage girls in India. We solved this problem by using Python programming.

The various nutrients like carbohydrates, proteins, fats, vitamins and minerals in different food items of various food groups like cereals and millets, grain legumes, green leafy vegetables, other vegetables, fruits, roots and tubers, condiments and spices-fresh, nuts and oil seeds, sugars, milk and milk products, are helpful for maintenance, growth, reproduction and health of human beings.

In this paper the various food items are taken as decision variables and constraints are designed corresponding to different nutrients. In the construction of constraints we have assumed that any intake of more than the minimum requirement of nutrients is not harmful to the human body. Here the objective is to find the optimum solution that is to find the quantity of food items that should be consumed to minimize the cost of diet which will fulfill the minimum requirement of nutrients for teenage girls.

Keywords: Nutrients, Diet, pulp, Lp_prob.solve(), Lp_prob.objective, pandas, Dataframe.

1. INTRODUCTION

Indian teenage girls in the age group 10-18 years, need a wide range of nutrients to perform various functions in their body and to lead a healthy life. Since girls derive all the nutrients they need through the diet they eat, their diet must be well balanced to provide all the vital nutrients in proper proportion [2]. Since teenage girls are in the high need of protein, fats, carbohydrates, vitamin A and C, iron, calcium, folate and also magnesium to have a good nutritional status which will keep them free from diseases like anemia, underweight, PCOS, etc. Hence their diet must have good accessibility to these nutrients through their regular diet which must be easily available to them. To find optimum nutritional benefits in low cost here we formulate the LPP (Diet Problem) in which we have to specify first, the four things which are,

1.1. Decision Variables:

The variables used to decide the output as decision variables. We have taken 49 vegetarian food items as decision variables (as shown in table 2) which Indian girls commonly use in their daily diet as a sustaining food. In this diet problem using LPP we have to find out how many grams of these food items are necessary for the daily requirement of nutrients for girls at minimum cost.

1.2. Objective Function:

The linear function which is to be optimized is called the objective function. In this problem, our main objective is to minimize the total cost of the daily diet of girls which will fulfill the minimum requirement of nutrients. Here we want the cost of food items purchased (as shown in table 2) to be as little as possible and at the same time, we also emphasize on the fact that these food items provide all the necessary nutrients required for the healthy life of girls. Some food items can be common for different nutrients. It is beneficial for cost cutting of diet.

1.3. Constraints:

These are the restrictions on the decision variables, i.e., food items should provide minimum nutritional requirements necessary for healthy life. There are many nutrients which are available in various food items, but for convenience, for teenage girls, we considered 10 nutrients which are commonly available in all types of food items and their minimum amounts which are required for sustaining a healthy life recommended by Indian Council of Medical Research (ICMR) are as shown in table 1[1].

Table 1. ICMR Recommended dietary allowances per day for Indians

Nutrients (gm/day)	Protein	Fats	Carbo-hydrates	Vit. A	Folate	Vit. C	Vit. D	Iron	Calcium	Magnesium
Girls (Body wt. 36.4 kg to 55.7 kg)	43.2	67.7	130	0.00086	0.000245	0.066	0.000015	0.03	0.00001	0.325

1.4 Non Negative Restrictions:

All the decision variables are restricted to be non negative.

Mathematically, the diet problem can be stated as follows,

$$Min\ Z = \sum_{j=1}^{n} c_j x_j \qquad \dots(1)$$

Subject to the constraints,

$$\sum_{j=1}^{n} a_{ij} x_j \geq b_i \quad ; i = 1, 2, \dots, m \qquad \dots(2)$$

$$x_j \geq 0 \quad ; j = 1, 2, \dots, n \qquad \dots(3)$$

Where x_j's are the quantities of food items, c_j's are the costs of food items per 100 gm, a_{ij}'s are amounts of nutrients per 100 gm of food items and b_i's are daily minimum nutritional requirements.

Our diet problem is to minimize the function (1) subject to the constraints (2) and non negative restrictions (3).
Here the diet problem is of minimization and constraints are of greater than or equal to type.We solved it using a computer-based method i.e., Python programming for planning an optimal menu with respect to the daily nutritional requirements of Indian teenage girls [6].
Following the formulation of LPP and attainment of an optimum solution of it, it is often desired to study the effect of changes in the different parameters of the problem on the current optimum solution. If slight changes are made in the parameters or the structure of a given LPP after its optimum solution has been attained then the analysis of such post-optimal problems can thus be termed as post-optimality analysis or sensitivity analysis [5].

2. FORMULATION OF DIET PROBLEM

Table 2 provides us the input data for a LPP with 49 decision variables indicating the amount of various food items to be consumed and 10 constraints determining minimum nutritional requirement [3][4].

Table 2. Amount of various nutrients contained in different vegetarian food groups-items with costs

Sr. No.	Food Groups		Cost in Rs/100 gm	Nutrients (gm / day)									
	(Food Items in 100 gm)			Protein	Fats	Carbo-hydrates	Calcium	Iron	Vit. A	Vit. B9 (Folate)	Vit. C	Vit. D2	Magnesium
1	Cereals, Grains And Products			43.2	67.7	130	0.000001	0.03	0.00086	0.000245	0.066	0.000015	0.325
	Wheat Flour	x1	5	12.1	1.7	69.4	0.048	0.0049	0.000284	0.00002922	0	0.0000152	0.125
	Wheat Semolina	x2	6.5	10.4	0.8	74.8	0.016	0.0016	0.000276	0.00002568	0	0.00000819	0.03789
	Rice Flacks	x3	4	6.6	1.2	77.3	0.02	0.02	0.00003361	0.00000846	0	0	0.07792
	Rice Puffed	x4	5	7.5	0.1	73.6	0.023	0.0066	0.00005046	0	0	0	0.06459
	Maize (Dry)	x5	2.5	11.1	3.6	1.5	0.01	0.0023	0.000893	0.00002581	0	0.0000336	0.145
2	Pulses And Legumes												
	Red Gram Dal	x6	12	22.3	1.7	57.6	0.073	0.0027	0.000484	0.000108	0	0.00000212	0.119
	Green Gram Dal	x7	11	24.5	1.2	59.9	0.075	0.0039	0.000619	0.00009211	0	0.00000205	0.155
	Bengal Gram Dal	x8	7.6	20.8	5.6	59.8	0.056	0.0053	0.000999	0.000182	0	0.00000175	0.118
	Black Gram Dal	x9	12	24.0	1.4	59.6	0.154	0.0038	0.000463	0.00008875	0	0.00000842	0.173
	Moth Bean	x10	14.8	23.6	1.1	56.5	0.202	0.0095	0.000622	0.000349	0	0.00000977	0.205
	Peas Green	x11	6	7.2	0.1	15.9	0.02	0.0015	0.001286	0.00005477	0.0384	0.00001521	0.04011
3	Leafy Vegetables												
	Spinach	x12	6	2.0	0.7	2.9	0.073	0.00114	0.009553	0.000142	0.03028	0.00000026	0.08697
	Fenugreek Leaves	x13	16	4.4	0.9	34	0.08	0.0006	0.012577	0.00007526	0.05825	0.00000236	0.06367
	Coriender leaves	x14	10	3.3	0.6	6.3	0.184	0.00142	0.013808	0.00005101	0.02387	0	0.07268
	Cabbage	x15	6	1.8	0.1	46	0.039	0.0008	0.000339	0.00004636	0.03325	0.00000021	0.01799
4	Roots And Tubers												
	Redish (Pink)	x16	6	0.6	0.3	6.8	0.05	0.00037	0.00001761	0.00002465	0.1763	0.00000004	0.02225
	Onion	x17	3	1.8	0.1	12.6	0.04	0.0012	0.00003104	0.00002968	0.01096	0.00000012	0.01516
	Garlik	x18	4.7	36.3	0.1	29.8	0.03	0.0012	0.00003048	0.00007882	0.1357	0.00000197	0.02578
	Ginger	x19	8	2.3	0.9	12.3	0.02	0.0035	0.00025	0.00001082	0.00543	0.0000031	0.03686

	Potato	x20	4	1.6	0.1	22.6	0.01	0.00048	0.000224	0.00001385	0.02641	0.00000019	0.02234
5	Condiments And Spices												
	Cumin Seeds	x21	32	8.7	15.0	36.6	1.08	0.0117	0.000553	0.00002779	0	0.0000121	0.442
	Black Papper	x22	120	11.5	6.8	49.2	0.46	0.0124	0.002219	0.00002189	0	0.00002568	0.196
	Turmeric	x23	14	6.3	5.1	69.4	0.15	0.0678	0.000427	0.00001386	0	0.00001867	0.26
6	Other Vegetables												
	French Beans (Country)	x24	11.5	1.7	0.1	4.5	0.05	0.00061	0.001501	0.00004745	0.01581	0.00000182	0.04301
	Pumpkin Fruit	x25	6	1.4	0.1	4.6	0.01	0.00044	0.001449	0.00002414	0.00804	0.0000014	0.01043
	Brinjal	x26	8	1.4	0.3	4	0.018	0.00038	0.000309	0.00003393	0.00209	0.00000104	0.021
	Cauliflower	x27	8	2.6	0.4	4	0.033	0.00123	0.00005048	0.00004595	0.04714	0.00000132	0.02308
	Tomato	x28	4	1.9	0.1	3.6	0.02	0.0018	0.000546	0.00001251	0.01641	0.0000011	0.01357
	Capsicum	x29	8	1.3	0.3	4.3	0.01	0.000567	0.002511	0.00005185	0.00123	0.0000007	0.01184
	Ladyfingers	x30	4	1.9	0.2	6.4	0.066	0.00035	0.001223	0.0006368	0.02251	0.00000746	0.0661
	Cucumber	x31	21	0.4	0.1	2.5	0.01	0.0006	0.000171	0.00001467	0.00621	0.00000136	0.01848
7	Fruits												
	Banana	x32	5	1.2	0.3	27.2	0.017	0.00036	0.00026	0.00001793	0.00806	0.0000002	0.03022
	Guava	x33	10	0.9	0.3	11.2	0.01	0.00027	0.000996	0.00002976	0.214	0.00000168	0.015261
	Sweet Lemon	x34	10	0.7	0.3	7.3	0.03	0.0007	0.00008513	0.00001538	0.04696	0.0000003	0.0089
	Orange	x35	18	0.7	0.2	10.9	0.026	0.00032	0.000675	0.00001946	0.04272	0.00000034	0.01105
	Grapes	x36	24	0.5	0.3	16.5	0.02	0.00052	0.000208	0.00000835	0.01710	0.00000359	0.00687
8	Nuts And Oil Seeds												
	Sesame Seeds	x37	25	18.3	43.3	25	1.45	0.0025	0.00005706	0.000131	0	0.00006274	0.372
	Groundnuts	x38	13	25.3	40.1	26.1	0.09	0.0025	0.0000828	0.00009087	0	0.0000071	0.197
	Mustard seeds	x39	20	20	39.7	23.8	0.49	0.0079	0.00675	0.00009488	0	0.00003179	0.266
	Dry Coconut	x40	24	6.8	62.3	18.4	0.4	0.0078	0.000176	0.00002427	0	0	0.09721
9	Milk And Milk Products												
	Milk(Buffalo)	x41	6	4.3	6.5	5	0.21	0.0002	0.00008087	0.00000857	0.00237	0.00000163	0.01005
	Paneer	x42	49	13.4	23	7.9	0.48	0	0.000194	0.00009331	0	0.00000013	0.02662
	curd	x43	15	3.1	4	3	0.149	0.0002	0	0	0	0	0

	Cheese	x44	57	24.1	25.1	6.3	0.79	0.0021	0	0	0	0	0
10	Fats And Edible Oils												
	Cooking Oil	x45	21	0	100	0	0	0	0	0	0	0	0
	Ghee	x46	53	0	100	0	0	0	0	0	0	0	0
	Butter	x47	54	0	81	0	0	0	0	0	0	0	0
11	Sugars												
	Jaggery	x48	6	0.4	0.1	95	0.08	0.00264	0.00001849	0.0000144	0	0.00000047	0.115
	Honey	x49	38	0.3	0	79.5	0.005	0.000696	0	0	0	0	0

*The prices of food items are found from the grocery shop of Pune (MS, India) in the month of September 2022 i.e. in the end of rainy season and starting of winter season. Here we have taken seasonal leafy vegetable-s*pinach* and fruits - *Orange, Lime Sweet and Guava, etc.*

3. DEFINING DIET PROBLEM OF INDIAN GIRLS USING PYTHON PROGRAMMING

One of the easiest and one of the most used tools to code up a linear optimization problem in Python is using the PuLP library. PuLP is a free open source software written in Python. It is used to describe optimization problems as mathematical models. PuLP can then call any of numerous external LP solvers (CBC, GLPK, CPLEX, Gurobi etc) to solve this model and then use python commands to manipulate and display the solution. We install the library PuLP required for solver in our Python environment. [7][8]

The Python program for diet problem of indian girls is as given below:

```
# import the library pulp as p
import pulp as p

# Create a LP Minimization problem
Lp_prob = p.LpProblem('Problem', p.LpMinimize)

# Create problem Variables
x1 = p.LpVariable("x1", lowBound = 0)
x2 = p.LpVariable("x2", lowBound = 0)
x3 = p.LpVariable("x3", lowBound = 0)
x4 = p.LpVariable("x4", lowBound = 0)
x5 = p.LpVariable("x5", lowBound = 0)
x6 = p.LpVariable("x6", lowBound = 0)
x7 = p.LpVariable("x7", lowBound = 0)
x8 = p.LpVariable("x8", lowBound = 0)
x9 = p.LpVariable("x9", lowBound = 0)
x10 = p.LpVariable("x10", lowBound = 0)
x11 = p.LpVariable("x11", lowBound = 0)
x12 = p.LpVariable("x12", lowBound = 0)
x13 = p.LpVariable("x13", lowBound = 0)
x14 = p.LpVariable("x14", lowBound = 0)
x15 = p.LpVariable("x15", lowBound = 0)
x16 = p.LpVariable("x16", lowBound = 0)
x17 = p.LpVariable("x17", lowBound = 0)
x18 = p.LpVariable("x18", lowBound = 0)
x19 = p.LpVariable("x19", lowBound = 0)
x20 = p.LpVariable("x20", lowBound = 0)
x21 = p.LpVariable("x21", lowBound = 0)
x22 = p.LpVariable("x22", lowBound = 0)
x23 = p.LpVariable("x23", lowBound = 0)
x24 = p.LpVariable("x24", lowBound = 0)
x25 = p.LpVariable("x25", lowBound = 0)
x26 = p.LpVariable("x26", lowBound = 0)
x27 = p.LpVariable("x27", lowBound = 0)
x28 = p.LpVariable("x28", lowBound = 0)
x29 = p.LpVariable("x29", lowBound = 0)
```

```
x30 = p.LpVariable("x30", lowBound = 0)
x31 = p.LpVariable("x31", lowBound = 0)
x32 = p.LpVariable("x32", lowBound = 0)
x33 = p.LpVariable("x33", lowBound = 0)
x34 = p.LpVariable("x34", lowBound = 0)
x35 = p.LpVariable("x35", lowBound = 0)
x36 = p.LpVariable("x36", lowBound = 0)
x37 = p.LpVariable("x37", lowBound = 0)
x38 = p.LpVariable("x38", lowBound = 0)
x39 = p.LpVariable("x39", lowBound = 0)
x40 = p.LpVariable("x40", lowBound = 0)
x41 = p.LpVariable("x41", lowBound = 0)
x42 = p.LpVariable("x42", lowBound = 0)
x43 = p.LpVariable("x43", lowBound = 0)
x44 = p.LpVariable("x44", lowBound = 0)
x45 = p.LpVariable("x45", lowBound = 0)
x46 = p.LpVariable("x46", lowBound = 0)
x47 = p.LpVariable("x47", lowBound = 0)
x48 = p.LpVariable("x48", lowBound = 0)
x49 = p.LpVariable("x49", lowBound = 0)

# Objective Function
Lp_prob+=5*x1+6.5*x2+4*x3+5*x4+2.5*x5+12*x6+11*x7+7.6*x8+12*x9+14.8*x10+6*x11+6*x12+16*x13+
10*x14+6*x15+6*x16+3*x17+4.7*x18+8*x19+4*x20+32*x21+120*x22+14*x23+11.5*x24+6*x25+8*x26+8*
x27+4*x28+8*x29+4*x30+21*x31+5*x32+10*x33+10*x34+18*x35+24*x36+25*x37+13*x38+20*x39+24*x4
0          +6*x41+49*x42+15*x43+57*x44+21*x45+53*x46+54*x47+6*x48+38*x49

# Constraints:
# Constraint corresponding to Protein:
Lp_prob+=12.1*x1+10.4*x2+6.6*x3+7.5*x4+11.1*x5+22.3*x6+24.0*x7+20.8*x8+20.4*x9+23.6*x10+7.2*x11
+2.0*x12+4.4*x13+3.3*x14+1.8*x15+0.6*x16+1.8*x17+36.3*x18+2.3*x19+1.6*x20+8.7*x21+11.5*x22+6.3*x
23+1.7*x24+1.4*x25+1.4*x26+2.6*x27+1.9*x28+1.3*x29+0.2*x30+1.9*x31+0.4*x32+0.9*x33+0.7*x34+0.7*x
35+0.5*x36+18.3*x37+25.3*x38+20*x39+6.8*x40+4.3*x41+13.4*x42+3.1*x43+24.1*x44+0*x45+0*x46+0*x4
7+0.4*x48+0.3*x49 >= 43.2

# Constraint corresponding to Fats:
Lp_prob+=1.7*x1+0.8*x2+1.2*x3+0.1*x4+3.6*x5+1.7*x6+1.2*x7+5.6*x8+1.4*x9+1.1*x10+0.1*x11+0.7*x12
+0.9*x13+0.6*x14+0.1*x15+0.3*x16+0.1*x17+0.1*x18+0.9*x19+0.1*x20+15.0*x21+6.8*x22+5.1*x23+0.1*x2
4+0.1*x25+0.3*x26+0.4*x27+0.1*x28+0.3*x29+0.2*x30+0.1*x31+0.3*x32+0.3*x33+0.3*x34+0.2*x35+0.3*x3
6+43.3*x37+40.1*x38+39.7*x39+62.3*x40+6.5*x41+23*x42+4*x43+25.1*x44+100*x45+100*x46+81*x47+0.
1*x48+0*x49 >= 67.7

# Constraint corresponding to Carbohydrates:
Lp_prob+=69.4*x1+74.8*x2+77.3*x3+73.6*x4+1.5*x5+57.6*x6+59.9*x7+59.8*x8+59.6*x9+56.5*x10+15.9*x
11+2.9*x12+34*x13+6.3*x14+46*x15+6.8*x16+12.6*x17+29.8*x18+12.3*x19+22.6*x20+36.6*x21+49.2*x22
+69.4*x23+4.5*x24+4.6*x25+4*x26+4*x27+3.6*x28+4.3*x29+6.4*x30+2.5*x31+27.2*x32+11.2*x33+7.3*x34
+10.9*x35+16.5*x36+25*x37+26.1*x38+23.8*x39+18.4*x40+5*x41+7.9*x42+3*x43+6.3*x44+0*x45+0*x46+
0*x47+95*x48+79.5*x49 >=130

# Constraint corresponding to Calcium:
Lp_prob+=0.048*x1+0.016*x2+0.020*x3+0.023*x4+0.01*x5+0.073*x6+0.075*x7+0.056*x8+0.154*x9+0.202*
x10+0.02*x11+0.073*x12+0.08*x13+0.184*x14+0.039*x15+0.05*x16+0.04*x17+0.03*x18+0.02*x19+0.01*x2
0+1.08*x21+0.46*x22+0.15*x23+0.05*x24+0.01*x25+0.018*x26+0.033*x27+0.02*x28+0.01*x29+0.066*x30+
0.01*x31+0.017*x32+0.01*x33+0.03*x34+0.026*x35+0.02*x36+1.45*x37+0.09*x38+0.49*x39+0.4*x40+0.21
*x41+0.48*x42+0.149*x43+0.79*x44+0*x45+0*x46+0*x47+0.08*x48+0.005*x49 >= 0.000001

# Constraint corresponding to Iron:
Lp_prob+=0.0049*x1+0.0016*x2+0.02*x3+0.0066*x4+0.0023*x5+0.0027*x6+0.0039*x7+0.0053*x8+0.0038*
x9+0.0095*x10+0.0015*x11+0.00114*x12+0.0006*x13+0.00142*x14+0.0008*x15+0.00037*x16+0.0012*x17+
0.0012*x18+0.0035*x19+0.00048*x20+0.0117*x21+0.0124*x22+0.0678*x23+0.00061*x24+0.00044*x25+0.00
038*x26+0.00123*x27+0.0018*x28+0.000567*x29+0.00035*x30+0.0006*x31+0.00036*x32+0.00027*x33+0.0
007*x34+0.00032*x35+0.00052*x36+0.0025*x37+0.0025*x38+0.0079*x39+0.0078*x40+0.0002*x41+0*x42+
0.0002*x43+0.0021*x44+0*x45+0*x46+0*x47+0.00264*x48+0.000696*x49 >=0.03
```

```
# Constraint corresponding to Vitamin A:
Lp_prob+=0.000284*x1+0.000276*x2+0.00003361*x3+0.00005046*x4+0.000893*x5+0.000484*x6+0.000619*
x7+0.000999*x8+0.000463*x9+0.000622*x10+0.001286*x11+0.009553*x12+0.012577*x13+0.013808*x14+0.
000339*x15+0.00001761*x16+0.00003104*x17+0.00003048*x18+0.00025*x19+0.000224*x20+0.000553*x21
+0.002219*x22+0.000427*x23+0.001501*x24+0.001449*x25+0.000309*x26+0.00005048*x27+0.000546*x28+
0.002511*x29+0.001223*x30+0.000171*x31+0.00026*x32+0.000996*x33+0.00008513*x34+0.000675*x35+0.
000208*x36+0.00005706*x37+0.0000828*x38+0.00675*x39+0.000176*x40+0.00008087*x41+0.000194*x42+
0*x43+0*x44+0*x45+0*x46+0*x47+0.00001849*x48+0*x49 >=0.00086

# Constraint corresponding to Vitamin D2:
Lp_prob+=0.0000152*x1+0.00000819*x2+0*x3+0*x4+0.0000336*x5+0.00000212*x6+0.00000205*x7+0.0000
0175*x8+0.00000842*x9+0.00000977*x10+0.00001521*x11+0.00000026*x12+0.00000236*x13+0*x14+0.000
00021*x15+0.0000004*x16+0.00000012*x17+0.00000197*x18+0.0000031*x19+0.00000019*x20+0.0000121*
x21+0.00002568*x22+0.00001867*x23+0.00000182*x24+0.0000014*x25+0.00000104*x26+0.00000132*x27+
0.000011*x28+0.0000007*x29+0.00000746*x30+0.00000136*x31+0.0000002*x32+0.000000168*x33+0.00000
03*x34+0.00000034*x35+0.00000359*x36+0.00006274*x37+0.0000071*x38+0.00003179*x39+0*x40+0.0000
0163*x41+0.00000013*x42+0*x43+0*x44+0*x45+0*x46+0*x47+0.00000047*x48+0*x49 >=0.000015

## Constraint corresponding to Vitamin B9:
Lp_prob+=0.00002922*x1+0.00002568*x2+0.00000846*x3+0*x4+0.00002581*x5+0.000108*x6+0.00009211*
x7+0.000182*x8+0.00008875*x9+0.000349*x10+0.00005477*x11+0.000142*x12+0.00007526*x13+0.0000510
1*x14+0.00004636*x15+0.00002465*x16+0.00002968*x17+0.00007882*x18+0.00001082*x19+0.00001385*x
20+0.00002779*x21+0.00002189*x22+0.00001386*x23+0.00004745*x24+0.00002414*x25+0.00003393*x26+
0.00004595*x27+0.00001251*x28+0.00005185*x29+0.0006368*x30+0.00001467*x31+0.00001793*x32+0.000
02976*x33+0.00001538*x34+0.00001946*x35+0.00000835*x36+0.000131*x37+0.00009087*x38+0.00009488
*x39+0.00002427*x40+0.00000857*x41+0.00009331*x42+0*x43+0*x44+0*x45+0*x46+0*x47+0.0000144*x4
8+0*x49 >=0.000245

# Constraint corresponding to Vitamin C:
Lp_prob+=0*x1+0*x2+0*x3+0*x4+0*x5+0*x6+0*x7+0*x8+0*x9+0*x10+0.0384*x11+0.03028*x12+0.05825*
x13+0.02387*x14+0.03325*x15+0.1763*x16+0.01096*x17+0.1357*x18+0.00543*x19+0.02641*x20+0*x21+0
*x22+0*x23+0.01581*x24+0.00804*x25+0.00209*x26+0.04714*x27+0.01641*x28+0.00123*x29+0.02251*x30
+0.00621*x31+0.00806*x32+0.214*x33+0.04696*x34+0.04272*x35+0.01710*x36+0*x37+0*x38+0*x39+0*x4
0+0.00237*x41+0*x42+0*x43+0*x44+0*x45+0*x46+0*x47+0*x48+0*x49 >=0.066
# Constraint corresponding to Magnesium:
Lp_prob+=0.125*x1+0.03789*x2+0.07792*x3+0.06459*x4+0.145*x5+0.119*x6+0.155*x7+0.118*x8+0.173*x
9+0.205*x10+0.04011*x11+0.08697*x12+0.06367*x13+0.07268*x14+0.01799*x15+0.02225*x16+0.01516*x1
7+0.02578*x18+0.03686*x19+0.02234*x20+0.442*x21+0.196*x22+0.26*x23+0.04301*x24+0.01043*x25+0.0
21*x26+0.02308*x27+0.01357*x28+0.01184*x29+0.0661*x30+0.01848*x31+0.03022*x32+0.015261*x33+0.0
089*x34+0.01105*x35+0.00687*x36+0.372*x37+0.197*x38+0.266*x39+0.09721*x40+0.01005*x41+0.02662*
x42+0*x43+0*x44+0*x45+0*x46+0*x47+0.115*x48+0*x49 >=0.325

status = Lp_prob.solve() # Solver

# Sensitivity Analysis:
import pandas as pd
Lp_prob.solve()

print("Model Status:{}".format(p.LpStatus[Lp_prob.status]))
print("Objective = ", p.value(Lp_prob.objective))

for v in Lp_prob.variables():
    print(v.name,"=", v.varValue)

o = [{'Name':name,'Constraint':c,'shadow price':c.pi,'slack': c.slack} for name, c in Lp_prob.constraints.items()]

print(pd.DataFrame(o))
```

4. SOLUTION OF DIET PROBLEM OF INDIAN GIRLS

Table 3: Output of Python Program of diet problem of Indian girls.

```
Model
Status:Optimal
Objective =
25.29066561700
0002
x1 = 0.0
x10 = 0.0
x11 = 0.0
x12 = 0.0
x13 = 0.0
x14 = 0.0
x15 = 0.0
x16 = 0.0
x17 = 0.0
x18 =
0.54232861
x19 = 0.0
x2 = 0.0
x20 = 0.0
x21 = 0.0
x22 = 0.0
x23 = 0.0
x24 = 0.0
x25 = 0.0
x26 = 0.0
x27 = 0.0
x28 = 0.0
x29 = 0.0
x3 = 1.4276576
x30 =
0.24737143
x31 = 0.0
x32 = 0.0
x33 = 0.0
x34 = 0.0
x35 = 0.0
x36 = 0.0
x37 = 0.0
x38 = 0.0
x39 = 0.0
x4 = 0.0
x40 = 0.0
x41 = 0.0
x42 = 0.0
x43 = 0.0
x44 = 0.0
x45 =
0.61329118
x46 = 0.0
x47 = 0.0
x48 = 0.0
x49 = 0.0
x5 = 1.2649961
x6 = 0.0
x7 = 0.0
x8 = 0.0
x9 = 0.0
```

Here we get resultant optimum solution which gives the minimum cost (z) Rs. 25.29067 of food items satisfying minimum nutritional requirement (constraints) with values of basic variables as $x_3 = 1.43$, $x_5 = 1.27$, $x_{18}=0.54$, $x_{30}=0.25$ and $x_{45}= 0.61$ as shown in table 3.. Here x_3, x_5, x_{18}, x_{30} and x_{45} indicate quantities in 100 grams of rice flakes, dry maize, garlik, ladies fingers and Cooking oils like groundnuts oil, coconut oil,etc.

Finding the optimum solution to a linear programming model is only the first step. After that we perform the sensitivity analysis of the optimum solution and the output is as shown in table 4.

Table 4: Sensitivity Analysis of Optimum solution of diet problem of Indian girls.

```
   Name                          Constraint                      shadow price     slack
 0  _C1  {x1: 12.1, x2: 10.4, x3: 6.6, x4: 7.5, x5: 11....       0.084295      -0.000000
 1  _C2  {x1: 1.7, x2: 0.8, x3: 1.2, x4: 0.1, x5: 3.6, ...       0.210000      -0.000000
 2  _C3  {x1: 69.4, x2: 74.8, x3: 77.3, x4: 73.6, x5: 1...       0.036426      -0.000000
 3  _C4  {x1: 0.048, x2: 0.016, x3: 0.02, x4: 0.023, x5...       0.000000      -0.073798
 4  _C5  {x1: 0.0049, x2: 0.0016, x3: 0.02, x4: 0.0066,...       0.000000      -0.002200
 5  _C6  {x1: 0.000284, x2: 0.000276, x3: 3.361e-05, x4...       0.000000      -0.000637
 6  _C7  {x1: 1.52e-05, x2: 8.19e-06, x5: 3.36e-05, x6:...       0.000000      -0.000000
 7  _C8  {x1: 2.922e-05, x2: 2.568e-05, x3: 8.46e-06, x...    5382.805400      -0.000000
 8  _C9  {x11: 0.0384, x12: 0.03028, x13: 0.05825, x14:...       0.000000      -0.013162
 9 _C10  {x1: 0.125, x2: 0.03789, x3: 0.07792, x4: 0.06...       4.239663      -0.000000
```

5. SENSITIVITY ANALYSIS

5.1. Interpreting Dual Values:

Dual values are the most basic form of sensitivity analysis information. The dual value for a variable i.e. Reduced cost is nonzero only when the variable's value is equal to its upper or lower bound at the optimal solution. This variable is called a *nonbasic* variable, and its value was driven to the bound during the optimization process. The reduced costs tell us how much the objective function coefficients (costs per 100 gm) can be increased or decreased before the optimal solution changes.

The dual value for a constraint i.e., shadow price is nonzero only when the constraint is equal to its bound as shown in the figure 3. This is called a *binding* constraint, and its value was driven to the bound during the optimization process. These shadow prices tell us how much the optimal solution can be increased or decreased if we change the right hand side values (Minimum Requirement) with one unit.[6]

5.2. Interpreting Range Information

In this diet problem, the dual values are *constant* over a range of possible changes in the objective function coefficients and the constraint right hand sides. For each decision variable, the report shows its coefficient in the objective function, and the amount by which this coefficient could be increased or decreased without changing the dual value (Allowable Increase and Allowable Decrease). For each constraint, the report shows the constraint on the right hand side, and the amount by which the RHS could be increased or decreased without changing the dual value (Allowable Increase and Allowable Decrease)[6].

The shadow/dual prices Lower limits (allowable decrease) of constraints and the slack variables are as shown in the table 4.

6. CONCLUSIONS

The optimum solution with food items rice flakes, dry maize, garlik, ladies fingers and Cooking oils like groundnuts oil, coconut oil,etc. from which the the teenage girls (in the age group 10 to 18 years) in India can have sufficient nutritional benefits necessary for sustaining the healthy life by spending **Rs. 25.29067** per day only. This diet may not be tasteful but there is no doubt about the optimality of the solution within the limitations we have formulated.

The costs of food items play a very important role in finding the optimum solution to the diet problem. It should be noted that as costs always vary according to locations and seasons, the optimum solution changes.

The Sensitivity Report details how changes in the coefficients of the objective function affect the solution and how changes in the constants on the right hand side of the constraints affect the solution.For each variable, we can calculate the range of values that the coefficient can take on by subtracting the allowable decrease from the coefficient or adding the allowable increase to the coefficient. The second part of the Sensitivity Report examines how changes to the right hand side of any constraint affects the optimal solution. A change to the constant on the right hand side of a constraint changes the size of the feasible region. Increasing the right hand side of any constraint with positive coefficients shifts the border matching the constraint up. Decreasing the right hand side of any constraint with positive coefficients shifts the border matching the constraints down. The shadow price indicates how the objective function will change when the constant on the right hand side is changed [9].

The sensitivity analysis for a nonbinding constraint, like Carbohydrates Vitamin A, Vitamin C and calcium, is different. After we get the optimal solution, changes to the right hand side do not affect the costs as long as the right hand side is not decreased too much. This means that the shadow price is Rs. 0.

REFERENCES

[1] https://fssai.gov.in/upload/advisories/2021/07/60f1798019f94Direction_RDA_16_07_2021.pdf

[2] B. Srilakshmi, " Food Science" , New age international Publications.

[3]C. Gopalan, B .V. Rama Sastri, S. C. Balsubramanian, "Nutritive Value of Indian Foods", National institute of Nutrition, (ICMR), Hyderabad (India).

[4] T. Longvah, R. Ananthan, K. Bhaskaracharya, Venkaiah, "Indian Food Consumption Table", National institute of Nutrition, (ICMR), Hyderabad (India).

[5] Kanti swarup, P.K. Gupta, Man Mohan, "Operations Research", Sultan Chand and Sons.

[6] Pratibha Lakhani, Pooja Manjre, "APPLICATION OF R PROGRAMMING IN SOLUTION AND SENSITIVITY ANALYSIS OF DIET PROBLEM OF INDIAN MEN", International Journal of Science and Research (IJSR) ISSN: 2319-7064 SJIF (2019): 7.583, Volume 10 Issue 1, 904-908, January 2021.

[7] https://coin-or.github.io/pulp/main/installing_pulp_at_home.htm

[8] https://www.geeksforgeeks.org/python-linear-programming-in-pulp/

[9]P. B. Lakhani, Dipali Dhavane, Pooja Manjre, "Solution and Sensitivity Analysis of Diet Problem of Sedentary and Moderate Working Women in India", Internal Journal of Creative Research Thoughts, Volume 8 Issue 2, ISSN: 2320-2882, February 23, 2020.