Day 57

DIY

Q1. Problem Statement: Linear Programming using PuLP

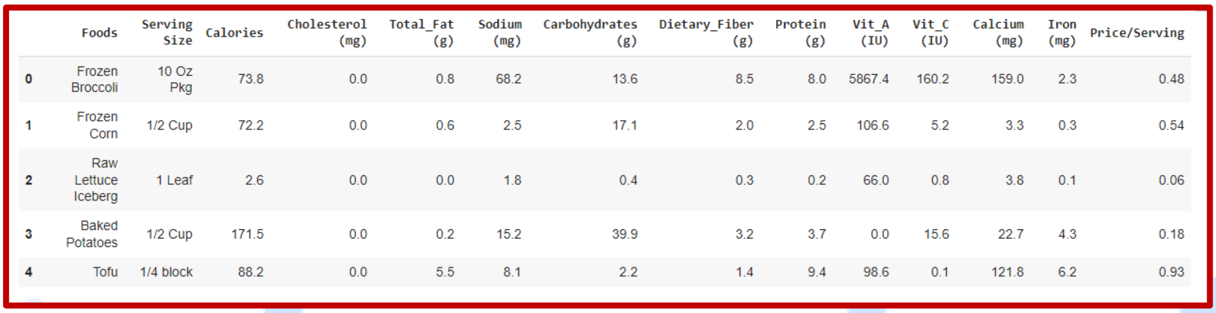
Suppose you are a dietician and are responsible to advise one of your customers on the best possible food plan he should follow in order to attain the optimum nutrition. However, there are some restrictions in terms of the budget and the variety of food that needs to be included in the diet plan. The cost should be minimal, at the same time, the nutritional value derived from the combination of different food items should be maximum, considering the maximum and minimum constraints given in the data.

Load the *'deit.csv'* dataset into a DataFrame and perform the following tasks:

1. Download and import PuLP and other required libraries
2. Formulate the optimization problem using `LpProblem()` method
3. Get a list of all the food items present in the DataFrame
4. Create a separate dictionary for each of these columns - *'Price/Serving', 'Calories', 'Cholesterol (mg)', 'Total_Fat (g)', 'Sodium (mg)', 'Carbohydrates (g)', 'Dietary_Fiber (g)', 'Protein (g)', 'Vit_A (IU)', 'Vit_C (IU)', 'Calcium (mg)', and 'Iron (mg)'* using `zip()` function
5. Create a dictionary called 'food_vars' to store the referenced variables for food items with the 'food' list
6. Add an objective function (`lpSum()`) to the LPP

7. Add the maximum and minimum constraints for each of the nutritional values present in the DataFrame
8. Run the solver algorithm to solve the problem
9. Print the variables after the solution
10. Based on the results, prepare an optimal diet plan

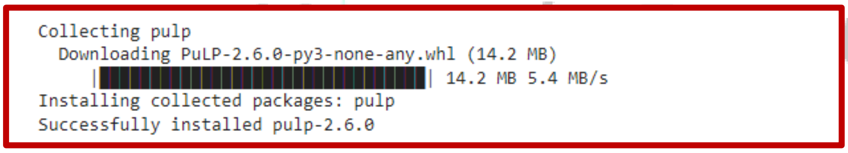
Dataset:



	Foods	Serving Size	Calories	Cholesterol (mg)	Total_Fat (g)	Sodium (mg)	Carbohydrates (g)	Dietary_Fiber (g)	Protein (g)	Vit_A (IU)	Vit_C (IU)	Calcium (mg)	Iron (mg)	Price/Serving
0	Frozen Broccoli	10 Oz Pkg	73.8	0.0	0.8	68.2	13.6	8.5	8.0	5867.4	160.2	159.0	2.3	0.48
1	Frozen Corn	1/2 Cup	72.2	0.0	0.6	2.5	17.1	2.0	2.5	106.6	5.2	3.3	0.3	0.54
2	Raw Lettuce Iceberg	1 Leaf	2.6	0.0	0.0	1.8	0.4	0.3	0.2	66.0	0.8	3.8	0.1	0.06
3	Baked Potatoes	1/2 Cup	171.5	0.0	0.2	15.2	39.9	3.2	3.7	0.0	15.6	22.7	4.3	0.18
4	Tofu	1/4 block	88.2	0.0	5.5	8.1	2.2	1.4	9.4	98.6	0.1	121.8	6.2	0.93

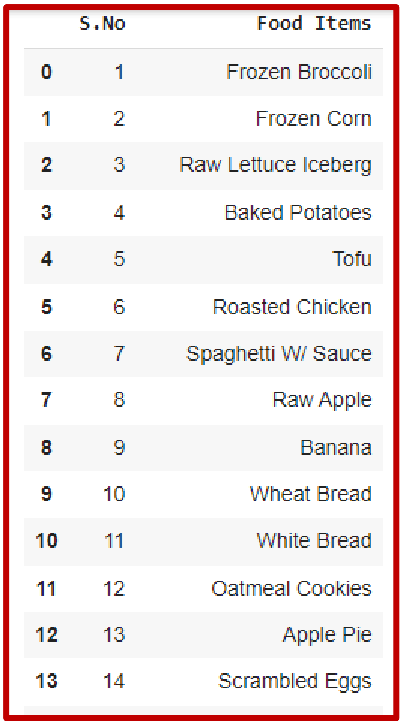
Sample Output:

1. Download and import PuLP and other required libraries



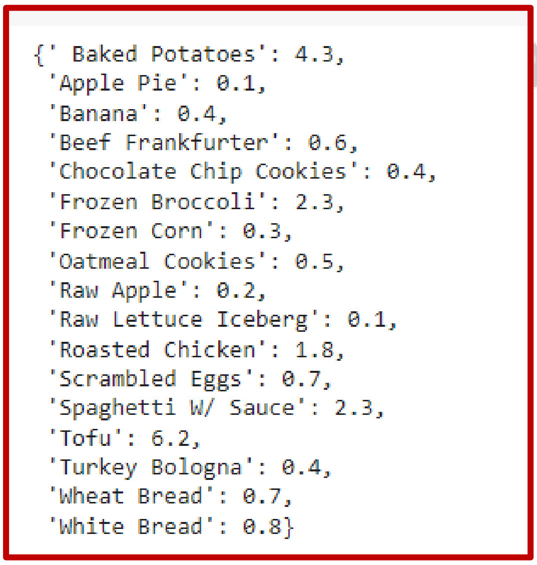
```
Collecting pulp
  Downloading PuLP-2.6.0-py3-none-any.whl (14.2 MB)
     |████████████████████████████████| 14.2 MB 5.4 MB/s
Installing collected packages: pulp
Successfully installed pulp-2.6.0
```

2. Formulate the optimization problem using `LpProblem()` method
3. Get a list of all the food items present in the DataFrame



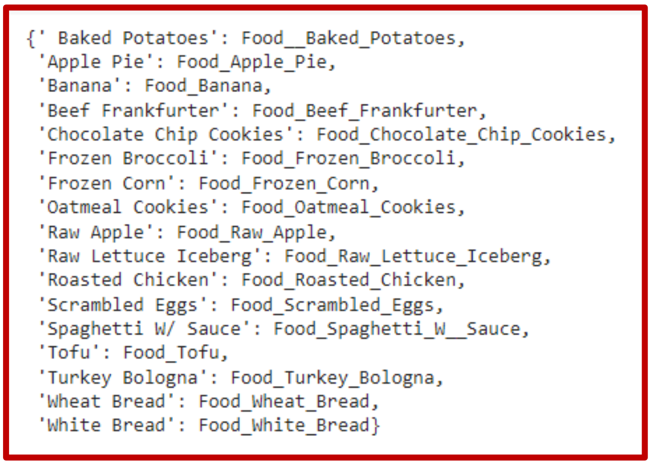
	S.No	Food Items
0	1	Frozen Broccoli
1	2	Frozen Corn
2	3	Raw Lettuce Iceberg
3	4	Baked Potatoes
4	5	Tofu
5	6	Roasted Chicken
6	7	Spaghetti W/ Sauce
7	8	Raw Apple
8	9	Banana
9	10	Wheat Bread
10	11	White Bread
11	12	Oatmeal Cookies
12	13	Apple Pie
13	14	Scrambled Eggs

4. Create a separate dictionary for each of these columns - *'Price/Serving', 'Calories', 'Cholesterol (mg)', 'Total_Fat (g)', 'Sodium (mg)', 'Carbohydrates (g)', 'Dietary_Fiber (g)', 'Protein (g)', 'Vit_A (IU)', 'Vit_C (IU)', 'Calcium (mg)', and 'Iron (mg)'* using `zip()` function



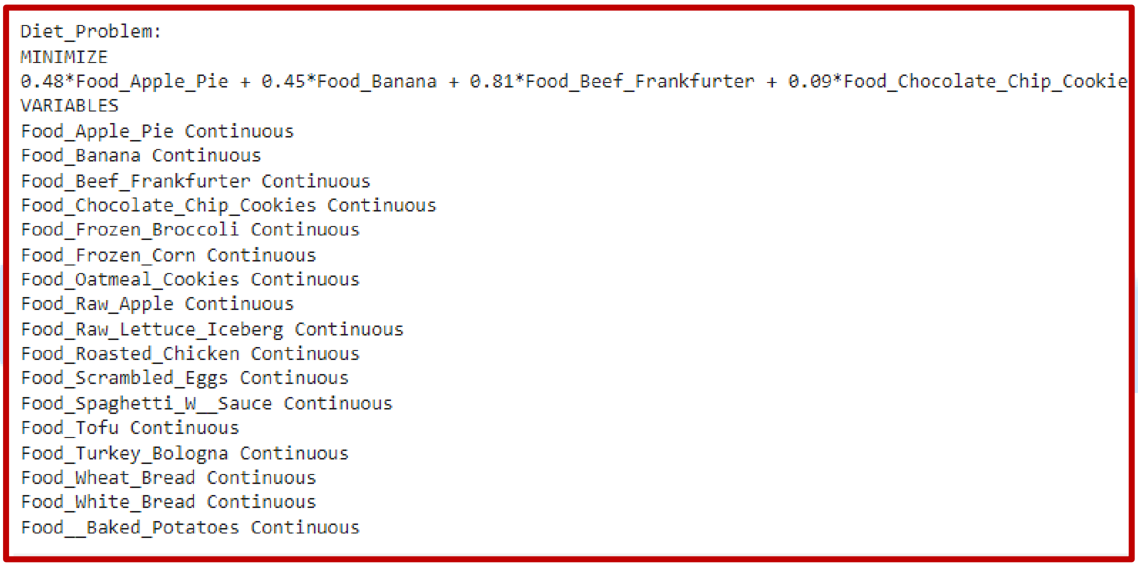
```
{'Baked Potatoes': 4.3,
 'Apple Pie': 0.1,
 'Banana': 0.4,
 'Beef Frankfurter': 0.6,
 'Chocolate Chip Cookies': 0.4,
 'Frozen Broccoli': 2.3,
 'Frozen Corn': 0.3,
 'Oatmeal Cookies': 0.5,
 'Raw Apple': 0.2,
 'Raw Lettuce Iceberg': 0.1,
 'Roasted Chicken': 1.8,
 'Scrambled Eggs': 0.7,
 'Spaghetti W/ Sauce': 2.3,
 'Tofu': 6.2,
 'Turkey Bologna': 0.4,
 'Wheat Bread': 0.7,
 'White Bread': 0.8}
```

5. Create a dictionary called 'food_vars' to store the referenced variables for food items with the 'food' list



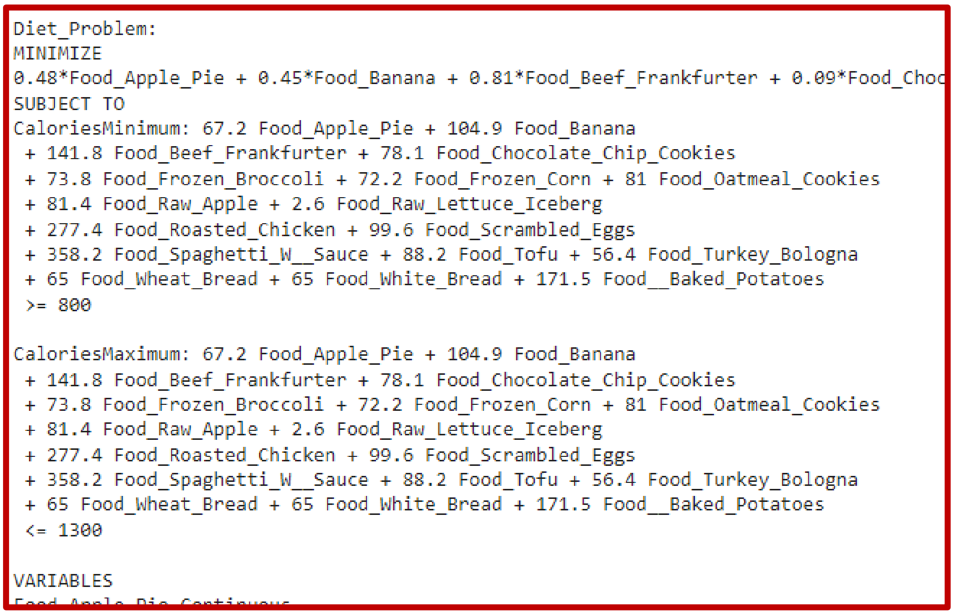
```
{' Baked Potatoes': Food__Baked_Potatoes,
 'Apple Pie': Food_Apple_Pie,
 'Banana': Food_Banana,
 'Beef Frankfurter': Food_Beef_Frankfurter,
 'Chocolate Chip Cookies': Food_Chocolate_Chip_Cookies,
 'Frozen Broccoli': Food_Frozen_Broccoli,
 'Frozen Corn': Food_Frozen_Corn,
 'Oatmeal Cookies': Food_Oatmeal_Cookies,
 'Raw Apple': Food_Raw_Apple,
 'Raw Lettuce Iceberg': Food_Raw_Lettuce_Iceberg,
 'Roasted Chicken': Food_Roasted_Chicken,
 'Scrambled Eggs': Food_Scrambled_Eggs,
 'Spaghetti W/ Sauce': Food_Spaghetti_W__Sauce,
 'Tofu': Food_Tofu,
 'Turkey Bologna': Food_Turkey_Bologna,
 'Wheat Bread': Food_Wheat_Bread,
 'White Bread': Food_White_Bread}
```

6. Add an objective function (`lpSum()`) to the LPP



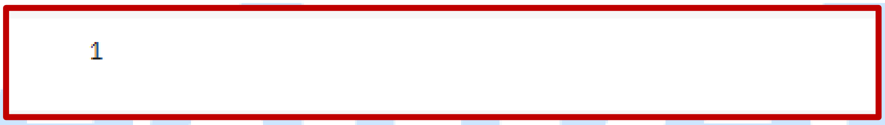
```
Diet_Problem:
MINIMIZE
0.48*Food_Apple_Pie + 0.45*Food_Banana + 0.81*Food_Beef_Frankfurter + 0.09*Food_Chocolate_Chip_Cookie
VARIABLES
Food_Apple_Pie Continuous
Food_Banana Continuous
Food_Beef_Frankfurter Continuous
Food_Chocolate_Chip_Cookies Continuous
Food_Frozen_Broccoli Continuous
Food_Frozen_Corn Continuous
Food_Oatmeal_Cookies Continuous
Food_Raw_Apple Continuous
Food_Raw_Lettuce_Iceberg Continuous
Food_Roasted_Chicken Continuous
Food_Scrambled_Eggs Continuous
Food_Spaghetti_W__Sauce Continuous
Food_Tofu Continuous
Food_Turkey_Bologna Continuous
Food_Wheat_Bread Continuous
Food_White_Bread Continuous
Food__Baked_Potatoes Continuous
```

7. Add the maximum and minimum constraints for each of the nutritional values present in the DataFrame

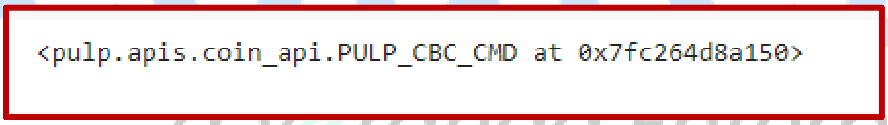


```
Diet_Problem:
MINIMIZE
0.48*Food_Apple_Pie + 0.45*Food_Banana + 0.81*Food_Beef_Frankfurter + 0.09*Food_Choc
SUBJECT TO
CaloriesMinimum: 67.2 Food_Apple_Pie + 104.9 Food_Banana
 + 141.8 Food_Beef_Frankfurter + 78.1 Food_Chocolate_Chip_Cookies
 + 73.8 Food_Frozen_Broccoli + 72.2 Food_Frozen_Corn + 81 Food_Oatmeal_Cookies
 + 81.4 Food_Raw_Apple + 2.6 Food_Raw_Lettuce_Iceberg
 + 277.4 Food_Roasted_Chicken + 99.6 Food_Scrambled_Eggs
 + 358.2 Food_Spaghetti_W__Sauce + 88.2 Food_Tofu + 56.4 Food_Turkey_Bologna
 + 65 Food_Wheat_Bread + 65 Food_White_Bread + 171.5 Food__Baked_Potatoes
 >= 800

CaloriesMaximum: 67.2 Food_Apple_Pie + 104.9 Food_Banana
 + 141.8 Food_Beef_Frankfurter + 78.1 Food_Chocolate_Chip_Cookies
 + 73.8 Food_Frozen_Broccoli + 72.2 Food_Frozen_Corn + 81 Food_Oatmeal_Cookies
 + 81.4 Food_Raw_Apple + 2.6 Food_Raw_Lettuce_Iceberg
 + 277.4 Food_Roasted_Chicken + 99.6 Food_Scrambled_Eggs
 + 358.2 Food_Spaghetti_W__Sauce + 88.2 Food_Tofu + 56.4 Food_Turkey_Bologna
 + 65 Food_Wheat_Bread + 65 Food_White_Bread + 171.5 Food__Baked_Potatoes
 <= 1300

VARIABLES
Food_Apple_Pie Continuous
```

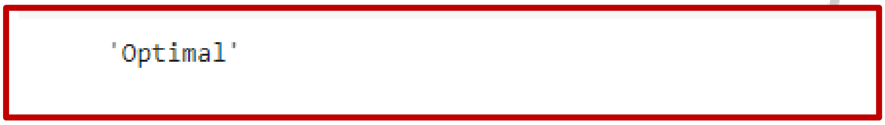
8. Run the solver algorithm to solve the problem



```
1
```

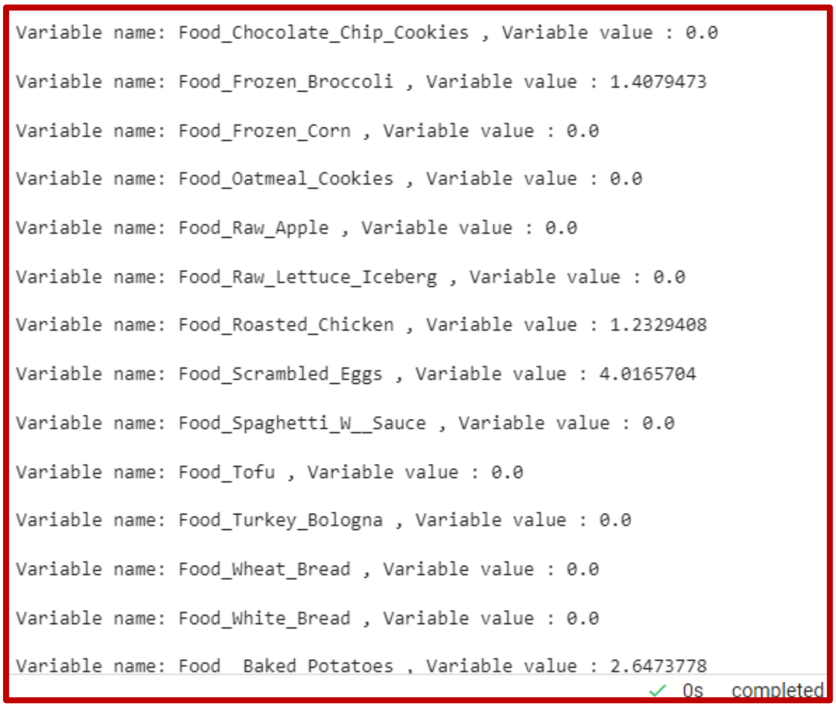


```
<pulp.apis.coin_api.PULP_CBC_CMD at 0x7fc264d8a150>
```



```
'Optimal'
```

9. Print the variables after the solution



```
Variable name: Food_Chocolate_Chip_Cookies , Variable value : 0.0

Variable name: Food_Frozen_Broccoli , Variable value : 1.4079473

Variable name: Food_Frozen_Corn , Variable value : 0.0

Variable name: Food_Oatmeal_Cookies , Variable value : 0.0

Variable name: Food_Raw_Apple , Variable value : 0.0

Variable name: Food_Raw_Lettuce_Iceberg , Variable value : 0.0

Variable name: Food_Roasted_Chicken , Variable value : 1.2329408

Variable name: Food_Scrambled_Eggs , Variable value : 4.0165704

Variable name: Food_Spaghetti_W__Sauce , Variable value : 0.0

Variable name: Food_Tofu , Variable value : 0.0

Variable name: Food_Turkey_Bologna , Variable value : 0.0

Variable name: Food_Wheat_Bread , Variable value : 0.0

Variable name: Food_White_Bread , Variable value : 0.0

Variable name: Food  Baked Potatoes , Variable value : 2.6473778
```

10.Based on the results, prepare an optimal diet plan