This is the Greenlist part

Created by Zanlai Hu on 2018/3/10. # Copyright © 2018 zanlaihu. All rights reserved.

Using read function to read the data, so that I can have a quick look.

[6]:
```
import pandas as pd

csv_file = "../input/ausnut1.csv"  # path to the downloaded csv file
ausdf = pd.read_csv(csv_file)
ausdf.head(5)
```

	Food ID	Survey ID	Food Name	Survey flag	Energy, with dietary fibre (kJ)	Energy, without dietary fibre (kJ)	Moisture (g)	Protein (g)	Total fat (g)	Available carbohydrates, with sugar alcohols (g)	Available carbohydrates, without sugar alcohol (g)	Starch (g)	Total sugars (g)	Added sugars (g)	Free sugars (g)	Dietary fibre (g)	Alcohol (g)	Ash (g)	Preformed vitamin A (retinol) (µg)	Beta-carotene (µg)	Provitamin A (b-carotene equivalents) (µg)
0	1	31103001	Beef, extract, bonox	NaN	401	401	56.6	16.6	0.2	6.5	6.5	6.5	0.0	0.0	0.0	0.0	0.0	19.8	0	0	0
1	2	31302001	Basil, dried	NaN	1079	774	10.0	18.2	5.5	15.5	15.5	15.5	0.0	0.0	0.0	38.2	0.0	15.5	0	27135	27334
2	3	31302002	Cardamom, seeds, ground	NaN	1333	1109	8.3	10.8	6.7	40.5	40.5	31.0	9.5	0.0	0.0	28.0	0.0	5.8	0	0	0
3	4	31302003	Chilli (chili) powder	NaN	1441	1167	7.8	12.3	16.8	20.5	20.5	10.4	10.1	0.0	0.0	34.2	0.0	11.8	0	15000	17790
4	5	31302004	Chilli (chili), dried, ground	NaN	1573	1355	6.6	14.1	16.8	29.6	29.6	19.2	10.4	0.0	0.0	27.2	0.0	10.0	0	15000	15000

There are many useless columns in the dataset. Firstly, I will leave only useful columns, Which are 'Food ID', 'Food Name', 'Energy, with dietary fibre (kJ)', 'Protein (g)', 'Added sugars (g)'

[7]:
```
columns_to_keep = ['Food ID',
                   'Food Name',
                   'Energy, with dietary fibre (kJ)',
                   'Protein (g)',
                   'Added sugars (g)'
                  ]

newdf = ausdf[columns_to_keep]
newdf.head(5)
```

	Food ID	Food Name	Energy, with dietary fibre (kJ)	Protein (g)	Added sugars (g)
0	1	Beef, extract, bonox	401	16.6	0.0
1	2	Basil, dried	1079	18.2	0.0
2	3	Cardamom, seeds, ground	1333	10.8	0.0
3	4	Chilli (chili) powder	1441	12.3	0.0
4	5	Chilli (chili), dried, ground	1573	14.1	0.0

Add two columns based on the value of Energy and Protein/added sugar.

[8]:
```
newdf = newdf.assign(protein_ratio=newdf['Protein (g)'] / newdf['Energy, with dietary fibre (kJ)'])
newdf = newdf.assign(addedsugar_ratio=newdf['Added sugars (g)'] / newdf['Energy, with dietary fibre (kJ)'])
newdf
```

	Food ID	Food Name	Energy, with dietary fibre (kJ)	Protein (g)	Added sugars (g)	protein_ratio	addedsugar_ratio
0	1	Beef, extract, bonox	401	16.6	0.0	0.041397	0.000000
1	2	Basil, dried	1079	18.2	0.0	0.016867	0.000000
2	3	Cardamom, seeds, ground	1333	10.8	0.0	0.008102	0.000000
3	4	Chilli (chili) powder	1441	12.3	0.0	0.008536	0.000000
4	5	Chilli (chili), dried, ground	1573	14.1	0.0	0.008964	0.000000
5	6	Cinnamon, dried, ground	1026	4.2	0.0	0.004094	0.000000
6	7	Cloves, dried, ground	1536	6.0	0.0	0.003906	0.000000
7	8	Coriander seed, dried, ground	1476	13.0	0.0	0.008808	0.000000
8	9	Cumin (cummin) seed, dried, ground	1915	18.4	0.0	0.009608	0.000000
9	10	Curry powder	1459	12.7	0.0	0.008705	0.000000
10	11	Fenugreek seed	1375	23.0	0.0	0.016727	0.000000
11	12	Ginger, dried, ground	1418	8.5	0.0	0.005994	0.000000
12	13	Herbs, mixed, dried	1210	11.1	0.0	0.009174	0.000000
13	14	Mustard powder	2289	29.4	0.0	0.012844	0.000000
14	15	Nutmeg, dried, ground	2119	6.4	0.0	0.003020	0.000000
15	16	Oregano or marjoram, dried	1096	9.0	0.0	0.008212	0.000000
16	17	Paprika, dry powder	1311	14.1	0.0	0.010755	0.000000
17	18	Pepper, ground, black or white	1208	10.7	0.0	0.008858	0.000000
18	19	Rosemary, dried	1405	9.5	0.0	0.006762	0.000000
19	20	Sage, dried	1320	10.6	0.0	0.008030	0.000000
20	21	Salt substitute, potassium chloride	0	0.0	0.0	NaN	NaN
21	22	Salt, cooking	0	0.0	0.0	NaN	NaN
22	23	Salt, flavoured	0	0.0	0.0	NaN	NaN
23	24	Salt, rock	0	0.0	0.0	NaN	NaN
24	25	Salt, sea	0	0.0	0.0	NaN	NaN
25	26	Salt, table, iodised	0	0.0	0.0	NaN	NaN
26	27	Salt, table, non-iodised	0	0.0	0.0	NaN	NaN
27	28	Salt, not further defined	0	0.0	0.0	NaN	NaN
28	29	Seasoning mix, chilli-based, for tacos	1010	5.9	11.2	0.005842	0.011089
29	30	Spice, mixed or all spice	1560	5.5	0.0	0.003526	0.000000
...	...	...	...	...	...	...	...
5710	5711	Tomato, fresh, boiled, microwaved or steamed, ...	304	1.2	0.0	0.003947	0.000000
5711	5712	Tomato, paste, with added salt	281	3.1	0.0	0.011032	0.000000
5712	5713	Tomato, paste, no added salt	298	3.1	0.0	0.010403	0.000000
5713	5714	Tomato, paste, not further defined	284	3.1	0.0	0.010915	0.000000
5714	5715	Tomato, puree, commercial	106	1.5	2.0	0.014151	0.018868
5715	5716	Tomato, sundried or semi-sundried	1101	11.2	0.0	0.010173	0.000000
5716	5717	Tomato, whole, canned in tomato juice, undrained	78	0.7	0.0	0.008974	0.000000
5717	5718	Tomato, whole, canned in tomato juice, drained	88	0.8	0.0	0.009091	0.000000
5718	5719	Tomato, whole, canned in tomato juice, boiled ...	94	0.8	0.0	0.008511	0.000000
5719	5720	Tomato, whole, canned in tomato juice, boiled ...	93	0.9	0.0	0.009677	0.000000
5720	5721	Turnip, white, peeled, fresh or frozen, raw	94	1.3	0.0	0.013830	0.000000
5721	5722	Turnip, white, peeled or unpeeled, fresh or fr...	115	1.6	0.0	0.013913	0.000000
5722	5723	Turnip, white, peeled or unpeeled, fresh or fr...	101	1.4	0.0	0.013861	0.000000
5723	5724	Vine leaf, grape, canned	243	4.3	0.0	0.017695	0.000000
5724	5725	Wasabi, root, raw	427	4.8	0.0	0.011241	0.000000
5725	5726	Water chestnut, peeled, canned, drained	219	0.6	0.0	0.002740	0.000000
5726	5727	Watercress, raw	110	2.9	0.0	0.026364	0.000000
5727	5728	Watercress, boiled, microwaved or steamed, dra...	130	3.4	0.0	0.026154	0.000000
5728	5729	Watercress, boiled, microwaved or steamed, dra...	320	3.3	0.0	0.010312	0.000000
5729	5730	Yam, wild harvested, cooked	453	3.2	0.0	0.007064	0.000000
5730	5731	Zucchini, golden, fresh or frozen, peeled or u...	78	2.2	0.0	0.028205	0.000000
5731	5732	Zucchini, golden, fresh or frozen, peeled or u...	93	2.6	0.0	0.027957	0.000000
5732	5733	Zucchini, green skin, fresh or frozen, peeled ...	65	0.9	0.0	0.013846	0.000000
5733	5734	Zucchini, green skin, fresh or frozen, peeled ...	92	1.3	0.0	0.014130	0.000000
5734	5735	Zucchini, green skin, fresh or frozen, peeled ...	299	1.3	0.0	0.004348	0.000000
5735	5736	Zucchini, green skin, fresh or frozen, peeled ...	352	1.2	0.0	0.003409	0.000000
5736	5737	Zucchini, green skin, fresh or frozen, peeled ...	352	1.2	0.0	0.003409	0.000000
5737	5738	Zucchini, green skin, fresh or frozen, peeled ...	347	1.2	0.0	0.003458	0.000000
5738	5739	Zucchini, green skin, fresh or frozen, peeled ...	77	1.1	0.0	0.014286	0.000000
5739	5740	Zucchini, green skin, fresh or frozen, peeled ...	293	1.0	0.0	0.003413	0.000000

5740 rows × 7 columns

I need to find the maximum value so that I can split them by different stars.

[9]:
```
by_protein = newdf.sort_values(by=["protein_ratio"], ascending=False)
by_protein.head(5)
```

	Food ID	Food Name	Energy, with dietary fibre (kJ)	Protein (g)	Added sugars (g)	protein_ratio	addedsugar_ratio
4686	4687	Amino acid or creatine powder	1666	98.0	0.0	0.058824	0.0
4483	4484	Shark (flake), baked, roasted, fried, grilled ...	615	35.9	0.0	0.058374	0.0
4503	4504	Stingray, wild caught, flesh, baked, roasted, ...	615	35.9	0.0	0.058374	0.0
4482	4483	Shark (flake), raw	449	26.2	0.0	0.058352	0.0
42	43	Gelatine, all types	1449	84.4	0.0	0.058247	0.0

[10]:
```
by_addsugar = newdf.sort_values(by=["addedsugar_ratio"], ascending=False)
by_addsugar.head(5)
```

	Food ID	Food Name	Energy, with dietary fibre (kJ)	Protein (g)	Added sugars (g)	protein_ratio	addedsugar_ratio
432	433	Water, bottled, with added sugar, vitamins & m...	70	0.0	4.4	0.0	0.062857
364	365	Soft drink, cola flavour, regular, with ice	126	0.0	7.9	0.0	0.062698
408	409	Soft drink, frozen, cola flavour, regular	203	0.0	12.7	0.0	0.062562
5038	5039	Syrup, agave, light & dark, liquid	1216	0.0	76.0	0.0	0.062500
5033	5034	Sugar, white, fruit sugar (fructose), granulat...	1600	0.0	100.0	0.0	0.062500

Now I calcualte the intervels.

[11]:
```
    max = 0.06
    #max protein ratio
    interval = max / 10
    j = 0
    for i in range(0,10):
        print("protein ratio from", j , "to", j + interval, "is", i + 1, "star")
        j = interval + j

    maxsu = 0.063
    #max added sugar ratio
    interval = max / 10
    j = 0
    for i in range(0,10):
        print("added sugar ratio from", j , "to", j + interval, "is", 5 - i, "star")
        j = interval + j
```

```
protein ratio from 0 to 0.006 is 1 star
protein ratio from 0.006 to 0.012 is 2 star
protein ratio from 0.012 to 0.018000000000000002 is 3 star
protein ratio from 0.018000000000000002 to 0.024 is 4 star
protein ratio from 0.024 to 0.03 is 5 star
protein ratio from 0.03 to 0.036 is 6 star
protein ratio from 0.036 to 0.041999999999999996 is 7 star
protein ratio from 0.041999999999999996 to 0.047999999999999994 is 8 star
protein ratio from 0.047999999999999994 to 0.053999999999999999 is 9 star
protein ratio from 0.05399999999999999 to 0.05999999999999999 is 10 star
added sugar ratio from 0 to 0.006 is 5 star
added sugar ratio from 0.006 to 0.012 is 4 star
added sugar ratio from 0.012 to 0.018000000000000002 is 3 star
added sugar ratio from 0.018000000000000002 to 0.024 is 2 star
added sugar ratio from 0.024 to 0.03 is 1 star
added sugar ratio from 0.03 to 0.036 is 0 star
added sugar ratio from 0.036 to 0.041999999999999996 is -1 star
added sugar ratio from 0.041999999999999996 to 0.047999999999999994 is -2 star
added sugar ratio from 0.047999999999999994 to 0.05399999999999999 is -3 star
added sugar ratio from 0.05399999999999999 to 0.05999999999999999 is -4 star
```

Then I create two columns based on the protein ratio and added sugar ratio.

[12]:
```
    def function1(a):
        if a < 0.006:
            return 1
        elif a < 0.012:
            return 2
        elif a < 0.018:
            return 3
        elif a < 0.024:
            return 4
        elif a < 0.03:
            return 5
        elif a < 0.036:
            return 6
        elif a < 0.042:
            return 7
        elif a < 0.048:
            return 8
        elif a < 0.054:
            return 9
        else:
            return 10

    def function2(a):
        if a < 0.006:
            return 10
        elif a < 0.012:
            return 9
        elif a < 0.018:
            return 8
        elif a < 0.024:
            return 7
        elif a < 0.03:
            return 6
        elif a < 0.036:
            return 5
        elif a < 0.042:
            return 4
        elif a < 0.048:
            return 3
        elif a < 0.054:
            return 2
        else:
            return 1
newdf['protein_star'] = newdf.apply(lambda x: function1(x.protein_ratio), axis=1)
newdf['addedsugar_star'] = newdf.apply(lambda x: function2(x.addedsugar_ratio), axis=1)
newdf.head(5)
```

	Food ID	Food Name	Energy, with dietary fibre (kJ)	Protein (g)	Added sugars (g)	protein_ratio	addedsugar_ratio	protein_star	addedsugar_star
0	1	Beef, extract, bonox	401	16.6	0.0	0.041397	0.0	7	10
1	2	Basil, dried	1079	18.2	0.0	0.016867	0.0	3	10
2	3	Cardamom, seeds, ground	1333	10.8	0.0	0.008102	0.0	2	10
3	4	Chilli (chili) powder	1441	12.3	0.0	0.008536	0.0	2	10
4	5	Chilli (chili), dried, ground	1573	14.1	0.0	0.008964	0.0	2	10

Finally, add a new column called aver_star, which is the average star of protein_star and addedsugar_star.

[13]:

```
def function3(a, b):
    c = (a + b)/2
    return c

newdf['aver_star'] = newdf.apply(lambda x: function3(x.protein_star, x.addedsugar_star), axis=1)
newdf.head(10)
```

	Food ID	Food Name	Energy, with dietary fibre (kJ)	Protein (g)	Added sugars (g)	protein_ratio	addedsugar_ratio	protein_star	addedsugar_star	aver_star
0	1	Beef, extract, bonox	401	16.6	0.0	0.041397	0.0	7	10	8.5
1	2	Basil, dried	1079	18.2	0.0	0.016867	0.0	3	10	6.5
2	3	Cardamom, seeds, ground	1333	10.8	0.0	0.008102	0.0	2	10	6.0
3	4	Chilli (chili) powder	1441	12.3	0.0	0.008536	0.0	2	10	6.0
4	5	Chilli (chili), dried, ground	1573	14.1	0.0	0.008964	0.0	2	10	6.0
5	6	Cinnamon, dried, ground	1026	4.2	0.0	0.004094	0.0	1	10	5.5
6	7	Cloves, dried, ground	1536	6.0	0.0	0.003906	0.0	1	10	5.5
7	8	Coriander seed, dried, ground	1476	13.0	0.0	0.008808	0.0	2	10	6.0
8	9	Cumin (cummin) seed, dried, ground	1915	18.4	0.0	0.009608	0.0	2	10	6.0
9	10	Curry powder	1459	12.7	0.0	0.008705	0.0	2	10	6.0

Sort it from top to bottom. Here is the top 10 healthy food.

[14]:

```
by_total = newdf.sort_values(by=["aver_star"], ascending=False)
by_total.head(10)
```

	Food ID	Food Name	Energy, with dietary fibre (kJ)	Protein (g)	Added sugars (g)	protein_ratio	addedsugar_ratio	protein_star	addedsugar_star	aver_star
3816	3817	Kangaroo, wild caught, flesh, cooked	469	25.4	0.0	0.054158	0.0	10	10	10.0
4686	4687	Amino acid or creatine powder	1666	98.0	0.0	0.058824	0.0	10	10	10.0
4567	4568	Prawn, king or medium, flesh, baked, roasted, ...	463	25.6	0.0	0.055292	0.0	10	10	10.0
4555	4556	Moreton bay bug, cooked, with or without fat	509	27.5	0.0	0.054028	0.0	10	10	10.0
2552	2553	Egg, chicken, white (albumen) only, fried, no ...	223	12.7	0.0	0.056951	0.0	10	10	10.0
4566	4567	Prawn, king or medium, raw (green)	371	20.5	0.0	0.055256	0.0	10	10	10.0
4552	4553	Lobster or crayfish, flesh, purchased steamed,...	407	22.0	0.0	0.054054	0.0	10	10	10.0
3808	3809	Kangaroo, rump, raw	373	20.3	0.0	0.054424	0.0	10	10	10.0
3788	3789	Buffalo, swamp, topside, raw	448	24.6	0.0	0.054911	0.0	10	10	10.0
4407	4408	Cod, Atlantic, flesh, raw	328	17.8	0.0	0.054268	0.0	10	10	10.0

[ ]: