Preference-Based Daily Diet Optimization

Group 6
Baihan Lin, Daehyun Kim, Xinyuan Liu, Yijun Ma

Problem

- Find an optimal diet for our group members with maximum happiness
  - Food dataset from Public Health England’s governmental website
  - Based on preference of different categories of food
  - Constraints: fulfilling nutrient requirements
- Method: linear programming
- Modeled as a modified knapsack problem

Background

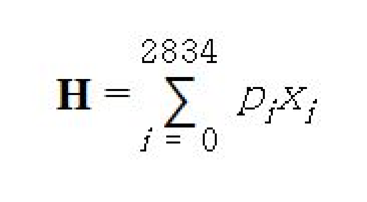
Related diet problem:

> Stifler Diet (1939): wanted to find the amount of each of 77 items of food to be eaten per day in order to at least meet the requirements of dietary allowances suggested by National Research Council with minimal cost.

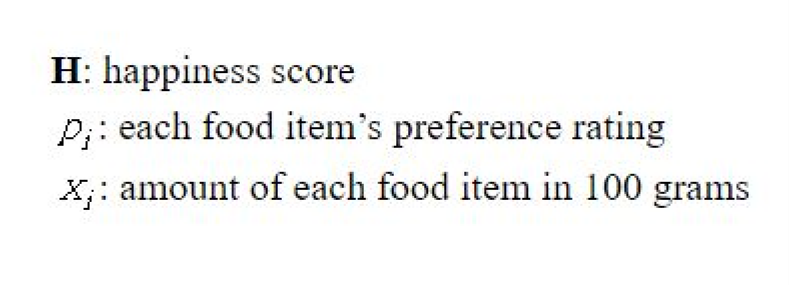
Model

Mathematical Model

Objective:



$$\mathbf{H} = \sum_{i=0}^{2834} p_i x_i$$



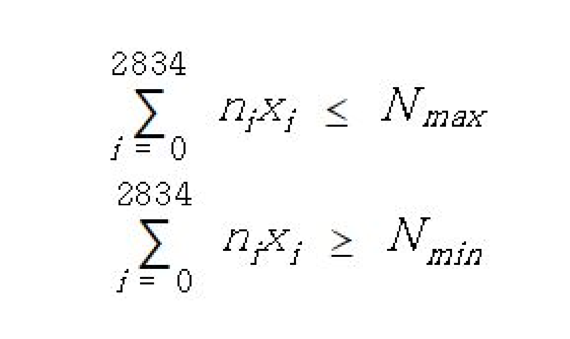
H: happiness score

p_i : each food item’s preference rating

x_i : amount of each food item in 100 grams

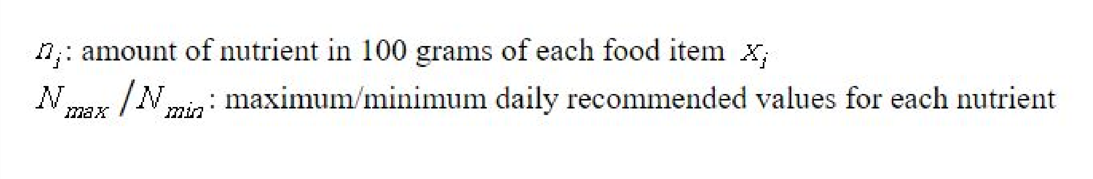
Model

Constraints:



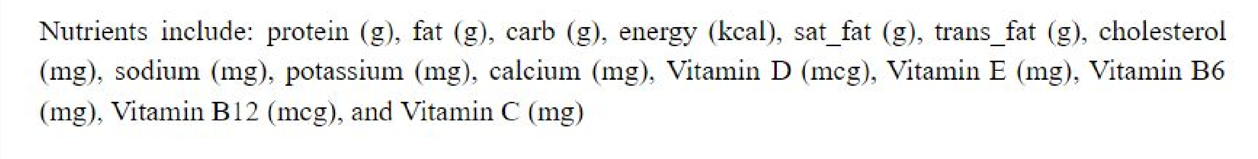
$$\sum_{i=0}^{2834} n_i x_i \leq N_{max}$$

$$\sum_{i=0}^{2834} n_i x_i \geq N_{min}$$



n_i: amount of nutrient in 100 grams of each food item x_i

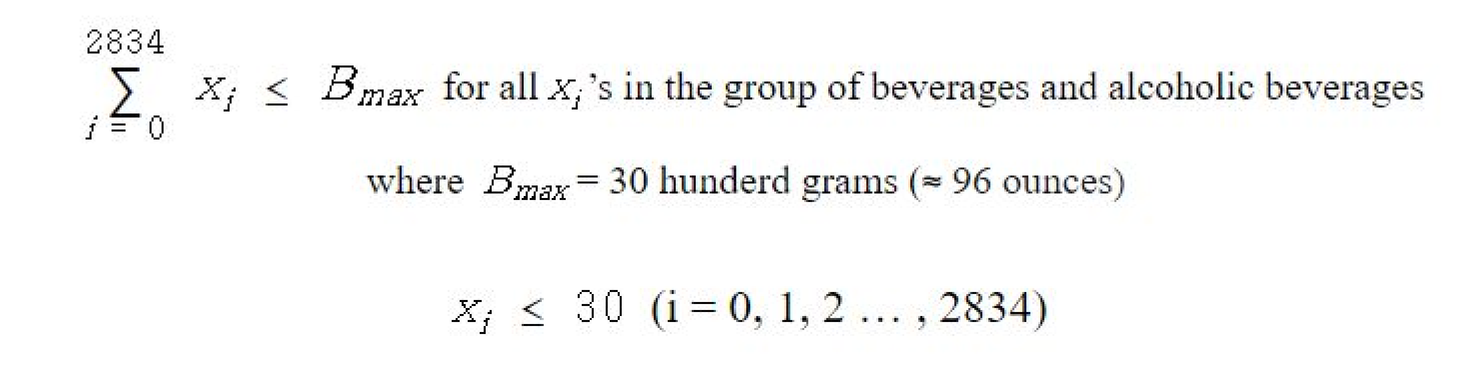
N_{max} / N_{min}: maximum/minimum daily recommended values for each nutrient



Nutrients include: protein (g), fat (g), carb (g), energy (kcal), sat_fat (g), trans_fat (g), cholesterol (mg), sodium (mg), potassium (mg), calcium (mg), Vitamin D (mcg), Vitamin E (mg), Vitamin B6 (mg), Vitamin B12 (mcg), and Vitamin C (mg)

Model

Constraints:



$\sum_{i=0}^{2834} x_i \le B_{max}$ for all x_i's in the group of beverages and alcoholic beverages

where B_{max} = 30 hunderd grams (≈ 96 ounces)

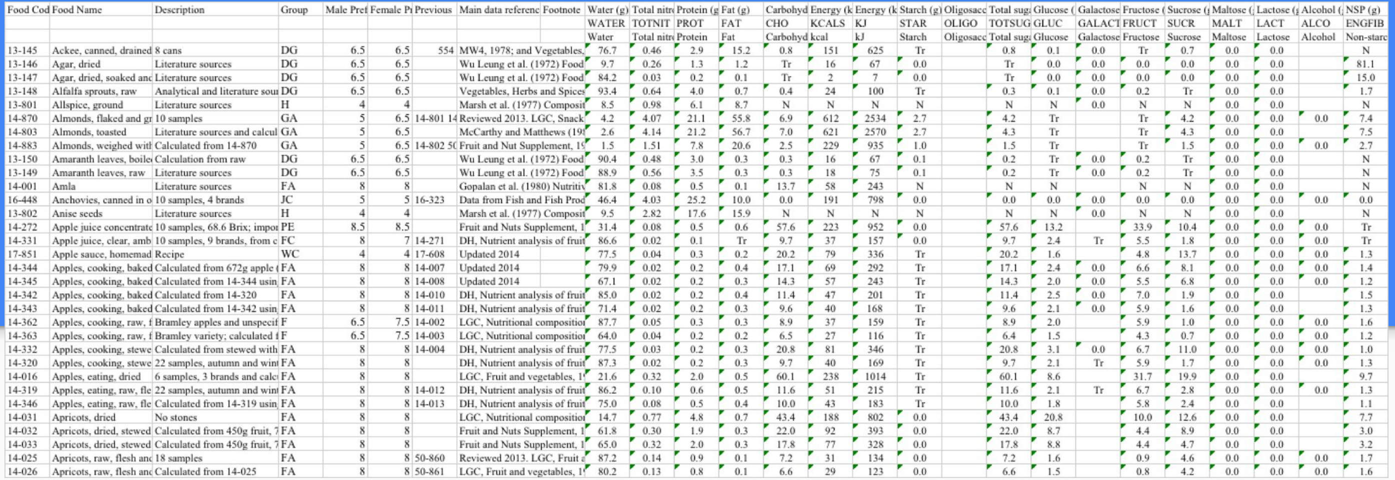
$x_i \le 30$ (i = 0, 1, 2 … , 2834)

Model

Implementation

1. Data Arrangement
2. Lpsolve Script Generation
3. Lpsolve Analysis

Model

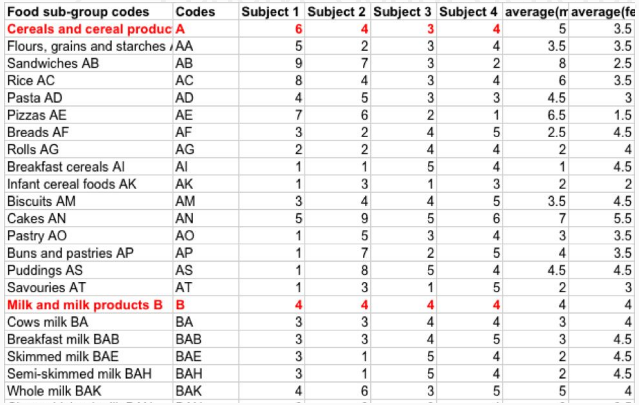


Food Cod	Food Name	Description	Group	Male Pref	Female Pr	Previous	Main data referenc	Footnote	Water (g)	Total nitr	Protein (g	Fat (g)	Carbohyd	Energy (k	Energy (k	Starch (g)	Oligosacc	Total sug	Glucose (	Galactose	Fructose (	Sucrose (	Maltose (	Lactose (	Alcohol (	NSP (g)
									WATER	TOTNIT	PROT	FAT	CHO	KCALS	KJ	STAR	OLIGO	TOTSUG	GLUC	GALACT	FRUCT	SUCR	MALT	LACT	ALCO	ENGFIB
									Water	Total nitr	Protein	Fat	Carbohyd	kcal	kJ	Starch	Oligosacc	Total sug	Glucose	Galactose	Fructose	Sucrose	Maltose	Lactose	Alcohol	Non-starc
13-145	Ackee, canned, drained	8 cans	DG	6.5	6.5	554	MW4, 1978; and Vegetables,		76.7	0.46	2.9	15.2	0.8	151	625	Tr		0.8	0.1	0.0	Tr	0.7	0.0	0.0		N
13-146	Agar, dried	Literature sources	DG	6.5	6.5		Wu Leung et al. (1972) Food		9.7	0.26	1.3	1.2	Tr	16	67	0.0		Tr	0.0	0.0	0.0	0.0	0.0	0.0		81.1
13-147	Agar, dried, soaked anc	Literature sources	DG	6.5	6.5		Wu Leung et al. (1972) Food		84.2	0.03	0.2	0.1	Tr	2	7	0.0		Tr	0.0	0.0	0.0	0.0	0.0	0.0		15.0
13-148	Alfalfa sprouts, raw	Analytical and literature sou	DG	6.5	6.5		Vegetables, Herbs and Spices		93.4	0.64	4.0	0.7	0.4	24	100	Tr		0.3	0.1	0.0	0.2	Tr	0.0	0.0		1.7
13-801	Allspice, ground	Literature sources	H	4	4		Marsh et al. (1977) Composit		8.5	0.98	6.1	8.7	N	N	N	N		N	N	0.0	N	N	0.0	0.0		N
14-870	Almonds, flaked and gr	10 samples	GA	5	6.5	14-801 14	Reviewed 2013. LGC, Snack		4.2	4.07	21.1	55.8	6.9	612	2534	2.7		4.2	Tr		Tr	4.2	0.0	0.0	0.0	7.4
14-803	Almonds, toasted	Literature sources and calcul	GA	5	6.5		McCarthy and Matthews (19		2.6	4.14	21.2	56.7	7.0	621	2570	2.7		4.3	Tr		Tr	4.3	0.0	0.0		7.5
14-883	Almonds, weighed witl	Calculated from 14-870	GA	5	6.5	14-802 5(	Fruit and Nut Supplement, 19		1.5	1.51	7.8	20.6	2.5	229	935	1.0		1.5	Tr		Tr	1.5	0.0	0.0	0.0	2.7
13-150	Amaranth leaves, boile	Calculation from raw	DG	6.5	6.5		Wu Leung et al. (1972) Food		90.4	0.48	3.0	0.3	0.3	16	67	0.1		0.2	Tr	0.0	0.2	Tr	0.0	0.0		N
13-149	Amaranth leaves, raw	Literature sources	DG	6.5	6.5		Wu Leung et al. (1972) Food		88.9	0.56	3.5	0.3	0.3	18	75	0.1		0.2	Tr	0.0	0.2	Tr	0.0	0.0		N
14-001	Amla	Literature sources	FA	8	8		Gopalan et al. (1980) Nutritiv		81.8	0.08	0.5	0.1	13.7	58	243	N		N	N		N	N	0.0	0.0		N
16-448	Anchovies, canned in o	10 samples, 4 brands	JC	5	5	16-323	Data from Fish and Fish Prod		46.4	4.03	25.2	10.0	0.0	191	798	0.0		0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
13-802	Anise seeds	Literature sources	H	4	4		Marsh et al. (1977) Composit		9.5	2.82	17.6	15.9	N	N	N	N		N	N	0.0	N	N	0.0	0.0		N
14-272	Apple juice concentrate	10 samples, 68.6 Brix; impo	PE	8.5	8.5		Fruit and Nuts Supplement, 1		31.4	0.08	0.5	0.6	57.6	223	952	0.0		57.6	13.2		33.9	10.4	0.0	0.0	0.0	Tr
14-331	Apple juice, clear, amb	10 samples, 9 brands, from c	FC	8	7	14-271	DH, Nutrient analysis of fruit		86.6	0.02	0.1	Tr	9.7	37	157	0.0		9.7	2.4	Tr	5.5	1.8	0.0	0.0	0.0	Tr
17-851	Apple sauce, homemad	Recipe	WC	4	4	17-608	Updated 2014		77.5	0.04	0.3	0.2	20.2	79	336	Tr		20.2	1.6		4.8	13.7	0.0	0.0	0.0	1.3
14-344	Apples, cooking, baked	Calculated from 672g apple (	FA	8	8	14-007	Updated 2014		79.9	0.02	0.2	0.4	17.1	69	292	Tr		17.1	2.4	0.0	6.6	8.1	0.0	0.0	0.0	1.4
14-345	Apples, cooking, baked	Calculated from 14-344 usin	FA	8	8	14-008	Updated 2014		67.1	0.02	0.2	0.3	14.3	57	243	Tr		14.3	2.0	0.0	5.5	6.8	0.0	0.0	0.0	1.2
14-342	Apples, cooking, baked	Calculated from 14-320	FA	8	8	14-010	DH, Nutrient analysis of fruit		85.0	0.02	0.2	0.4	11.4	47	201	Tr		11.4	2.5	0.0	7.0	1.9	0.0	0.0		1.5
14-343	Apples, cooking, baked	Calculated from 14-342 usin	FA	8	8	14-011	DH, Nutrient analysis of fruit		71.4	0.02	0.2	0.3	9.6	40	168	Tr		9.6	2.1	0.0	5.9	1.6	0.0	0.0		1.3
14-362	Apples, cooking, raw, f	Bramley apples and unspecif	F	6.5	7.5	14-002	LGC, Nutritional compositio		87.7	0.05	0.3	0.3	8.9	37	159	Tr		8.9	2.0		5.9	1.0	0.0	0.0	0.0	1.6
14-363	Apples, cooking, raw, f	Bramley variety; calculated f	F	6.5	7.5	14-003	LGC, Nutritional compositio		64.0	0.04	0.2	0.2	6.5	27	116	Tr		6.4	1.5		4.3	0.7	0.0	0.0	0.0	1.2
14-332	Apples, cooking, stewe	Calculated from stewed with	FA	8	8	14-004	DH, Nutrient analysis of fruit		77.5	0.03	0.2	0.3	20.8	81	346	Tr		20.8	3.1	0.0	6.7	11.0	0.0	0.0	0.0	1.0
14-320	Apples, cooking, stewe	22 samples, autumn and win	FA	8	8		DH, Nutrient analysis of fruit		87.3	0.02	0.2	0.3	9.7	40	169	Tr		9.7	2.1	Tr	5.9	1.7	0.0	0.0	0.0	1.3
14-016	Apples, eating, dried	6 samples, 3 brands and calc	FA	8	8		LGC, Fruit and vegetables, 1		21.6	0.32	2.0	0.5	60.1	238	1014	Tr		60.1	8.6		31.7	19.9	0.0	0.0		9.7
14-319	Apples, eating, raw, fle	22 samples, autumn and win	FA	8	8	14-012	DH, Nutrient analysis of fruit		86.2	0.10	0.6	0.5	11.6	51	215	Tr		11.6	2.1	Tr	6.7	2.8	0.0	0.0	0.0	1.3
14-346	Apples, eating, raw, fle	Calculated from 14-319 usin	FA	8	8	14-013	DH, Nutrient analysis of fruit		75.0	0.08	0.5	0.4	10.0	43	183	Tr		10.0	1.8		5.8	2.4	0.0	0.0		1.1
14-031	Apricots, dried	No stones	FA	8	8		LGC, Nutritional compositio		14.7	0.77	4.8	0.7	43.4	188	802	0.0		43.4	20.8		10.0	12.6	0.0	0.0		7.7
14-032	Apricots, dried, stewed	Calculated from 450g fruit, 7	FA	8	8		Fruit and Nuts Supplement, 1		61.8	0.30	1.9	0.3	22.0	92	393	0.0		22.0	8.7		4.4	8.9	0.0	0.0		3.0
14-033	Apricots, dried, stewed	Calculated from 450g fruit, 7	FA	8	8		Fruit and Nuts Supplement, 1		65.0	0.32	2.0	0.3	17.8	77	328	0.0		17.8	8.8		4.4	4.7	0.0	0.0		3.2
14-025	Apricots, raw, flesh anc	18 samples	FA	8	8	50-860	Reviewed 2013. LGC, Fruit a		87.2	0.14	0.9	0.1	7.2	31	134	0.0		7.2	1.6		0.9	4.6	0.0	0.0	0.0	1.7
14-026	Apricots, raw, flesh anc	Calculated from 14-025	FA	8	8	50-861	LGC, Fruit and vegetables, 1		80.2	0.13	0.8	0.1	6.6	29	123	0.0		6.6	1.5		0.8	4.2	0.0	0.0	0.0	1.6

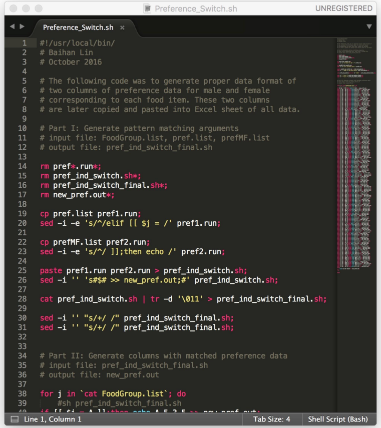
Implementation

1. Data Arrangement

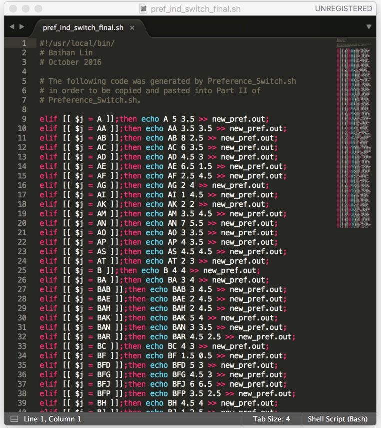
2. Unsolve Script Generation



Food sub-group codes	Codes	Subject 1	Subject 2	Subject 3	Subject 4	average(m	average(fe
Cereals and cereal produc	A	6	4	3	4	5	3.5
Flours, grains and starches /	AA	5	2	3	4	3.5	3.5
Sandwiches AB	AB	9	7	3	2	8	2.5
Rice AC	AC	8	4	3	4	6	3.5
Pasta AD	AD	4	5	3	3	4.5	3
Pizzas AE	AE	7	6	2	1	6.5	1.5
Breads AF	AF	3	2	4	5	2.5	4.5
Rolls AG	AG	2	2	4	4	2	4
Breakfast cereals AI	AI	1	1	5	4	1	4.5
Infant cereal foods AK	AK	1	3	1	3	2	2
Biscuits AM	AM	3	4	4	5	3.5	4.5
Cakes AN	AN	5	9	5	6	7	5.5
Pastry AO	AO	1	5	3	4	3	3.5
Buns and pastries AP	AP	1	7	2	5	4	3.5
Puddings AS	AS	1	8	5	4	4.5	4.5
Savouries AT	AT	1	3	1	5	2	3
Milk and milk products B	B	4	4	4	4	4	4
Cows milk BA	BA	3	3	4	4	3	4
Breakfast milk BAB	BAB	3	3	4	5	3	4.5
Skimmed milk BAE	BAE	3	1	5	4	2	4.5
Semi-skimmed milk BAH	BAH	3	1	5	4	2	4.5
Whole milk BAK	BAK	4	6	3	5	5	4



```bash
#!/usr/local/bin/
# Baihan Lin
# October 2016

# The following code was to generate proper data format of
# two columns of preference data for male and female
# corresponding to each food item. These two columns
# are later copied and pasted into Excel sheet of all data.

# Part I: Generate pattern matching arguments
# input file: FoodGroup.list, pref.list, prefMF.list
# output file: pref_ind_switch_final.sh

rm pref*.run*;
rm pref_ind_switch.sh*;
rm pref_ind_switch_final.sh*;
rm new_pref.out*;

cp pref.list pref1.run;
sed -i -e 's/^/elif [[ $j = /' pref1.run;

cp prefMF.list pref2.run;
sed -i -e 's/^/ ]];then echo /' pref2.run;

paste pref1.run pref2.run > pref_ind_switch.sh;
sed -i '' 's#$# >> new_pref.out;#' pref_ind_switch.sh;

cat pref_ind_switch.sh | tr -d '\011' > pref_ind_switch_final.sh;

sed -i '' "s/+/ /" pref_ind_switch_final.sh;
sed -i '' "s/+/ /" pref_ind_switch_final.sh;


# Part II: Generate columns with matched preference data
# input file: pref_ind_switch_final.sh
# output file: new_pref.out

for j in `cat FoodGroup.list`; do
    #sh pref_ind_switch_final.sh
```



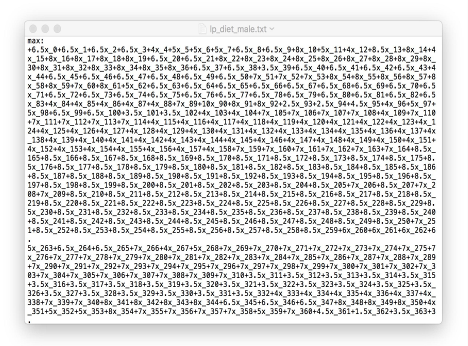
```bash
#!/usr/local/bin/
# Baihan Lin
# October 2016

# The following code was generated by Preference_Switch.sh
# in order to be copied and pasted into Part II of
# Preference_Switch.sh.

elif [[ $j = A ]];then echo A 5 3.5 >> new_pref.out;
elif [[ $j = AA ]];then echo AA 3.5 3.5 >> new_pref.out;
elif [[ $j = AB ]];then echo AB 8 2.5 >> new_pref.out;
elif [[ $j = AC ]];then echo AC 6 3.5 >> new_pref.out;
elif [[ $j = AD ]];then echo AD 4.5 3 >> new_pref.out;
elif [[ $j = AE ]];then echo AE 6.5 1.5 >> new_pref.out;
elif [[ $j = AF ]];then echo AF 2.5 4.5 >> new_pref.out;
elif [[ $j = AG ]];then echo AG 2 4 >> new_pref.out;
elif [[ $j = AI ]];then echo AI 1 4.5 >> new_pref.out;
elif [[ $j = AK ]];then echo AK 2 2 >> new_pref.out;
elif [[ $j = AM ]];then echo AM 3.5 4.5 >> new_pref.out;
elif [[ $j = AN ]];then echo AN 7 5.5 >> new_pref.out;
elif [[ $j = AO ]];then echo AO 3 3.5 >> new_pref.out;
elif [[ $j = AP ]];then echo AP 4 3.5 >> new_pref.out;
elif [[ $j = AS ]];then echo AS 4.5 4.5 >> new_pref.out;
elif [[ $j = AT ]];then echo AT 2 3 >> new_pref.out;
elif [[ $j = B ]];then echo B 4 4 >> new_pref.out;
elif [[ $j = BA ]];then echo BA 3 4 >> new_pref.out;
elif [[ $j = BAB ]];then echo BAB 3 4.5 >> new_pref.out;
elif [[ $j = BAE ]];then echo BAE 2 4.5 >> new_pref.out;
elif [[ $j = BAH ]];then echo BAH 2 4.5 >> new_pref.out;
elif [[ $j = BAK ]];then echo BAK 5 4 >> new_pref.out;
elif [[ $j = BAN ]];then echo BAN 3 3.5 >> new_pref.out;
elif [[ $j = BAR ]];then echo BAR 4.5 2.5 >> new_pref.out;
elif [[ $j = BC ]];then echo BC 4 3 >> new_pref.out;
elif [[ $j = BF ]];then echo BF 1.5 0.5 >> new_pref.out;
elif [[ $j = BFD ]];then echo BFD 5 3 >> new_pref.out;
elif [[ $j = BFG ]];then echo BFG 4.5 3 >> new_pref.out;
elif [[ $j = BFJ ]];then echo BFJ 6 6.5 >> new_pref.out;
elif [[ $j = BFP ]];then echo BFP 3.5 2.5 >> new_pref.out;
elif [[ $j = BH ]];then echo BH 4.5 4 >> new_pref.out;
```

Script: Bash
OS: Mac X
SW: iTerm2

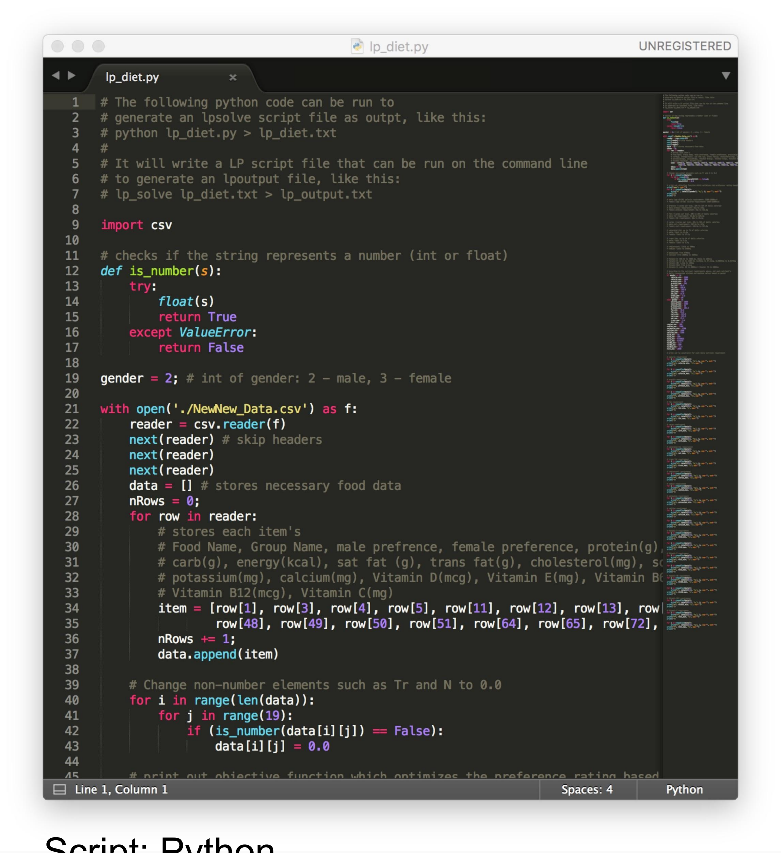
Model

Implementation

1. Data Arrangement
2. Lpsolve Script Generation
3. Lpsolve Analysis



```
max:
+6.5x_0+6.5x_1+6.5x_2+6.5x_3+4x_4+5x_5+5x_6+5x_7+6.5x_8+6.5x_9+8x_10+5x_11+4x_12+8.5x_13+8x_14+4
x_15+8x_16+8x_17+8x_18+8x_19+6.5x_20+6.5x_21+8x_22+8x_23+8x_24+8x_25+8x_26+8x_27+8x_28+8x_29+8x_
30+8x_31+8x_32+8x_33+8x_34+8x_35+8x_36+6.5x_37+6.5x_38+3.5x_39+6.5x_40+6.5x_41+6.5x_42+6.5x_43+4
x_44+6.5x_45+6.5x_46+6.5x_47+6.5x_48+6.5x_49+6.5x_50+7x_51+7x_52+7x_53+8x_54+8x_55+8x_56+8x_57+8
x_58+8x_59+7x_60+8x_61+5x_62+6.5x_63+6.5x_64+6.5x_65+6.5x_66+6.5x_67+6.5x_68+6.5x_69+6.5x_70+6.5
x_71+6.5x_72+6.5x_73+6.5x_74+6.5x_75+6.5x_76+6.5x_77+6.5x_78+6.5x_79+6.5x_80+6.5x_81+6.5x_82+6.5
x_83+4x_84+4x_85+4x_86+4x_87+4x_88+7x_89+10x_90+8x_91+8x_92+2.5x_93+2.5x_94+4.5x_95+4x_96+5x_97+
5x_98+6.5x_99+6.5x_100+3.5x_101+3.5x_102+4x_103+4x_104+7x_105+7x_106+7x_107+7x_108+4x_109+7x_110
+7x_111+7x_112+7x_113+7x_114+4x_115+4x_116+4x_117+4x_118+4x_119+4x_120+4x_121+4x_122+4x_123+4x_1
24+4x_125+4x_126+4x_127+4x_128+4x_129+4x_130+4x_131+4x_132+4x_133+4x_134+4x_135+4x_136+4x_137+4x
_138+4x_139+4x_140+4x_141+4x_142+4x_143+4x_144+4x_145+4x_146+4x_147+4x_148+4x_149+4x_150+4x_151+
4x_152+4x_153+4x_154+4x_155+4x_156+4x_157+4x_158+7x_159+7x_160+7x_161+7x_162+7x_163+7x_164+8.5x_
165+8.5x_166+8.5x_167+8.5x_168+8.5x_169+8.5x_170+8.5x_171+8.5x_172+8.5x_173+8.5x_174+8.5x_175+8.
5x_176+8.5x_177+8.5x_178+8.5x_179+8.5x_180+8.5x_181+8.5x_182+8.5x_183+8.5x_184+8.5x_185+8.5x_186
+8.5x_187+8.5x_188+8.5x_189+8.5x_190+8.5x_191+8.5x_192+8.5x_193+8.5x_194+8.5x_195+8.5x_196+8.5x_
197+8.5x_198+8.5x_199+8.5x_200+8.5x_201+8.5x_202+8.5x_203+8.5x_204+8.5x_205+7x_206+8.5x_207+7x_2
08+7x_209+8.5x_210+8.5x_211+8.5x_212+8.5x_213+8.5x_214+8.5x_215+8.5x_216+8.5x_217+8.5x_218+8.5x_
219+8.5x_220+8.5x_221+8.5x_222+8.5x_223+8.5x_224+8.5x_225+8.5x_226+8.5x_227+8.5x_228+8.5x_229+8.
5x_230+8.5x_231+8.5x_232+8.5x_233+8.5x_234+8.5x_235+8.5x_236+8.5x_237+8.5x_238+8.5x_239+8.5x_240
+8.5x_241+8.5x_242+8.5x_243+8.5x_244+8.5x_245+8.5x_246+8.5x_247+8.5x_248+8.5x_249+8.5x_250+7x_25
1+8.5x_252+8.5x_253+8.5x_254+8.5x_255+8.5x_256+8.5x_257+8.5x_258+8.5x_259+6x_260+6x_261+6x_262+6
.
5x_263+6.5x_264+6.5x_265+7x_266+4x_267+5x_268+7x_269+7x_270+7x_271+7x_272+7x_273+7x_274+7x_275+7
x_276+7x_277+7x_278+7x_279+7x_280+7x_281+7x_282+7x_283+7x_284+7x_285+7x_286+7x_287+7x_288+7x_289
+7x_290+7x_291+7x_292+7x_293+7x_294+7x_295+7x_296+7x_297+7x_298+7x_299+7x_300+7x_301+7x_302+7x_3
03+7x_304+7x_305+7x_306+7x_307+7x_308+7x_309+7x_310+3.5x_311+3.5x_312+3.5x_313+3.5x_314+3.5x_315
+3.5x_316+3.5x_317+3.5x_318+3.5x_319+3.5x_320+3.5x_321+3.5x_322+3.5x_323+3.5x_324+3.5x_325+3.5x_
326+3.5x_327+3.5x_328+3.5x_329+3.5x_330+3.5x_331+3.5x_332+4x_333+4x_334+4x_335+4x_336+4x_337+4x_
338+7x_339+7x_340+8x_341+8x_342+8x_343+8x_344+6.5x_345+6.5x_346+6.5x_347+8x_348+8x_349+8x_350+4x
_351+5x_352+5x_353+8x_354+7x_355+7x_356+7x_357+7x_358+5x_359+7x_360+4.5x_361+1.5x_362+3.5x_363+3
.
```



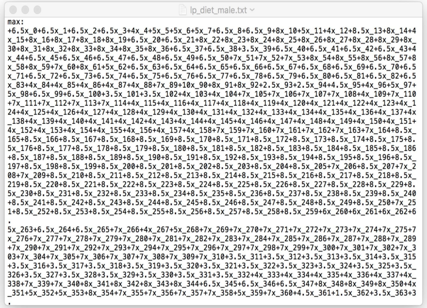
```python
# The following python code can be run to
# generate an lpsolve script file as outpt, like this:
# python lp_diet.py > lp_diet.txt
#
# It will write a LP script file that can be run on the command line
# to generate an lpoutput file, like this:
# lp_solve lp_diet.txt > lp_output.txt

import csv

# checks if the string represents a number (int or float)
def is_number(s):
    try:
        float(s)
        return True
    except ValueError:
        return False

gender = 2; # int of gender: 2 - male, 3 - female

with open('./NewNew_Data.csv') as f:
    reader = csv.reader(f)
    next(reader) # skip headers
    next(reader)
    next(reader)
    data = [] # stores necessary food data
    nRows = 0;
    for row in reader:
        # stores each item's
        # Food Name, Group Name, male prefrence, female preference, protein(g),
        # carb(g), energy(kcal), sat fat (g), trans fat(g), cholesterol(mg), sc
        # potassium(mg), calcium(mg), Vitamin D(mcg), Vitamin E(mg), Vitamin B6
        # Vitamin B12(mcg), Vitamin C(mg)
        item = [row[1], row[3], row[4], row[5], row[11], row[12], row[13], row
                row[48], row[49], row[50], row[51], row[64], row[65], row[72],
        nRows += 1;
        data.append(item)

    # Change non-number elements such as Tr and N to 0.0
    for i in range(len(data)):
        for j in range(19):
            if (is_number(data[i][j]) == False):
                data[i][j] = 0.0

    # print out objective function which optimizes the preference rating based
```

Script: Python
OS: Windows 10
SW: Eclipse, Python 3.5.1

Model

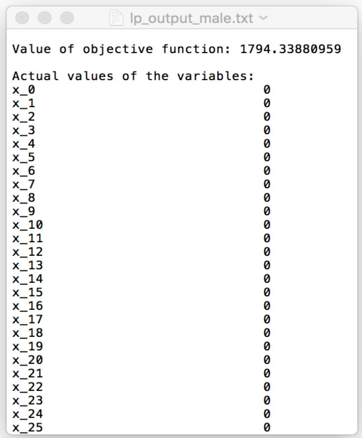
Implementation

1. Data Arrangement
2. Lpsolve Script Generation
3. Lpsolve Analysis



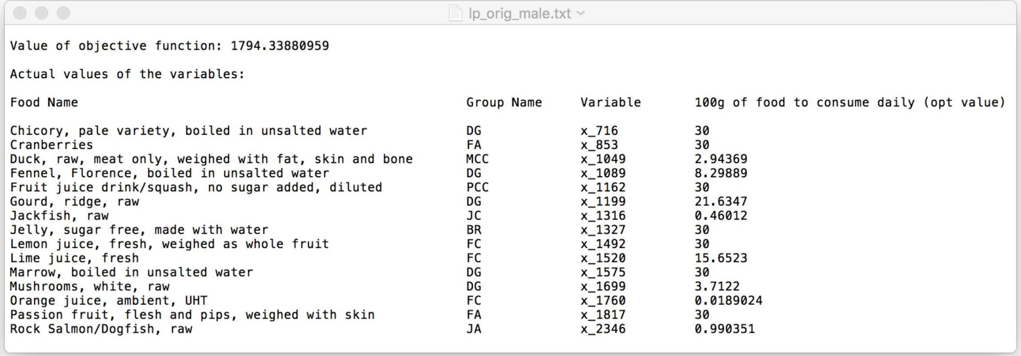
```
max:
+6.5x_0+6.5x_1+6.5x_2+6.5x_3+4x_4+5x_5+5x_6+5x_7+6.5x_8+6.5x_9+8x_10+5x_11+4x_12+8.5x_13+8x_14+4
x_15+8x_16+8x_17+8x_18+8x_19+6.5x_20+6.5x_21+8x_22+8x_23+8x_24+8x_25+8x_26+8x_27+8x_28+8x_29+8x_
30+8x_31+8x_32+8x_33+8x_34+8x_35+8x_36+6.5x_37+6.5x_38+3.5x_39+6.5x_40+6.5x_41+6.5x_42+6.5x_43+4
x_44+6.5x_45+6.5x_46+6.5x_47+6.5x_48+6.5x_49+6.5x_50+7x_51+7x_52+7x_53+8x_54+8x_55+8x_56+8x_57+8
x_58+8x_59+7x_60+8x_61+5x_62+6.5x_63+6.5x_64+6.5x_65+6.5x_66+6.5x_67+6.5x_68+6.5x_69+6.5x_70+6.5
x_71+6.5x_72+6.5x_73+6.5x_74+6.5x_75+6.5x_76+6.5x_77+6.5x_78+6.5x_79+6.5x_80+6.5x_81+6.5x_82+6.5
x_83+4x_84+4x_85+4x_86+4x_87+4x_88+7x_89+10x_90+8x_91+8x_92+2.5x_93+2.5x_94+4.5x_95+4x_96+5x_97+
5x_98+6.5x_99+6.5x_100+3.5x_101+3.5x_102+4x_103+4x_104+7x_105+7x_106+7x_107+7x_108+4x_109+7x_110
+7x_111+7x_112+7x_113+7x_114+4x_115+4x_116+4x_117+4x_118+4x_119+4x_120+4x_121+4x_122+4x_123+4x_1
24+4x_125+4x_126+4x_127+4x_128+4x_129+4x_130+4x_131+4x_132+4x_133+4x_134+4x_135+4x_136+4x_137+4x
_138+4x_139+4x_140+4x_141+4x_142+4x_143+4x_144+4x_145+4x_146+4x_147+4x_148+4x_149+4x_150+4x_151+
4x_152+4x_153+4x_154+4x_155+4x_156+4x_157+4x_158+7x_159+7x_160+7x_161+7x_162+7x_163+7x_164+8.5x_
165+8.5x_166+8.5x_167+8.5x_168+8.5x_169+8.5x_170+8.5x_171+8.5x_172+8.5x_173+8.5x_174+8.5x_175+8.
5x_176+8.5x_177+8.5x_178+8.5x_179+8.5x_180+8.5x_181+8.5x_182+8.5x_183+8.5x_184+8.5x_185+8.5x_186
+8.5x_187+8.5x_188+8.5x_189+8.5x_190+8.5x_191+8.5x_192+8.5x_193+8.5x_194+8.5x_195+8.5x_196+8.5x_
197+8.5x_198+8.5x_199+8.5x_200+8.5x_201+8.5x_202+8.5x_203+8.5x_204+8.5x_205+7x_206+8.5x_207+7x_2
08+7x_209+8.5x_210+8.5x_211+8.5x_212+8.5x_213+8.5x_214+8.5x_215+8.5x_216+8.5x_217+8.5x_218+8.5x_
219+8.5x_220+8.5x_221+8.5x_222+8.5x_223+8.5x_224+8.5x_225+8.5x_226+8.5x_227+8.5x_228+8.5x_229+8.
5x_230+8.5x_231+8.5x_232+8.5x_233+8.5x_234+8.5x_235+8.5x_236+8.5x_237+8.5x_238+8.5x_239+8.5x_240
+8.5x_241+8.5x_242+8.5x_243+8.5x_244+8.5x_245+8.5x_246+8.5x_247+8.5x_248+8.5x_249+8.5x_250+7x_25
1+8.5x_252+8.5x_253+8.5x_254+8.5x_255+8.5x_256+8.5x_257+8.5x_258+8.5x_259+6x_260+6x_261+6x_262+6
.
5x_263+6.5x_264+6.5x_265+7x_266+4x_267+5x_268+7x_269+7x_270+7x_271+7x_272+7x_273+7x_274+7x_275+7
x_276+7x_277+7x_278+7x_279+7x_280+7x_281+7x_282+7x_283+7x_284+7x_285+7x_286+7x_287+7x_288+7x_289
+7x_290+7x_291+7x_292+7x_293+7x_294+7x_295+7x_296+7x_297+7x_298+7x_299+7x_300+7x_301+7x_302+7x_3
03+7x_304+7x_305+7x_306+7x_307+7x_308+7x_309+7x_310+3.5x_311+3.5x_312+3.5x_313+3.5x_314+3.5x_315
+3.5x_316+3.5x_317+3.5x_318+3.5x_319+3.5x_320+3.5x_321+3.5x_322+3.5x_323+3.5x_324+3.5x_325+3.5x_
326+3.5x_327+3.5x_328+3.5x_329+3.5x_330+3.5x_331+3.5x_332+4x_333+4x_334+4x_335+4x_336+4x_337+4x_
338+7x_339+7x_340+8x_341+8x_342+8x_343+8x_344+6.5x_345+6.5x_346+6.5x_347+8x_348+8x_349+8x_350+4x
_351+5x_352+5x_353+8x_354+7x_355+7x_356+7x_357+7x_358+5x_359+7x_360+4.5x_361+1.5x_362+3.5x_363+3
.
```

Script: lpsolve
OS: Mac X
SW: lpsolve



```
Value of objective function: 1794.33880959

Actual values of the variables:
x_0                             0
x_1                             0
x_2                             0
x_3                             0
x_4                             0
x_5                             0
x_6                             0
x_7                             0
x_8                             0
x_9                             0
x_10                            0
x_11                            0
x_12                            0
x_13                            0
x_14                            0
x_15                            0
x_16                            0
x_17                            0
x_18                            0
x_19                            0
x_20                            0
x_21                            0
x_22                            0
x_23                            0
x_24                            0
x_25                            0
```

Model

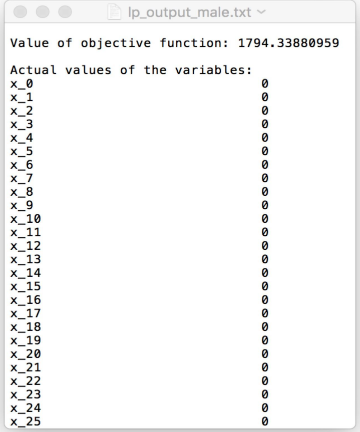


```
Value of objective function: 1794.33880959

Actual values of the variables:

Food Name                                                   Group Name     Variable       100g of food to consume daily (opt value)

Chicory, pale variety, boiled in unsalted water             DG             x_716          30
Cranberries                                                 FA             x_853          30
Duck, raw, meat only, weighed with fat, skin and bone       MCC            x_1049         2.94369
Fennel, Florence, boiled in unsalted water                  DG             x_1089         8.29889
Fruit juice drink/squash, no sugar added, diluted           PCC            x_1162         30
Gourd, ridge, raw                                           DG             x_1199         21.6347
Jackfish, raw                                               JC             x_1316         0.46012
Jelly, sugar free, made with water                          BR             x_1327         30
Lemon juice, fresh, weighed as whole fruit                  FC             x_1492         30
Lime juice, fresh                                           FC             x_1520         15.6523
Marrow, boiled in unsalted water                            DG             x_1575         30
Mushrooms, white, raw                                       DG             x_1699         3.7122
Orange juice, ambient, UHT                                  FC             x_1760         0.0189024
Passion fruit, flesh and pips, weighed with skin            FA             x_1817         30
Rock Salmon/Dogfish, raw                                    JA             x_2346         0.990351
```

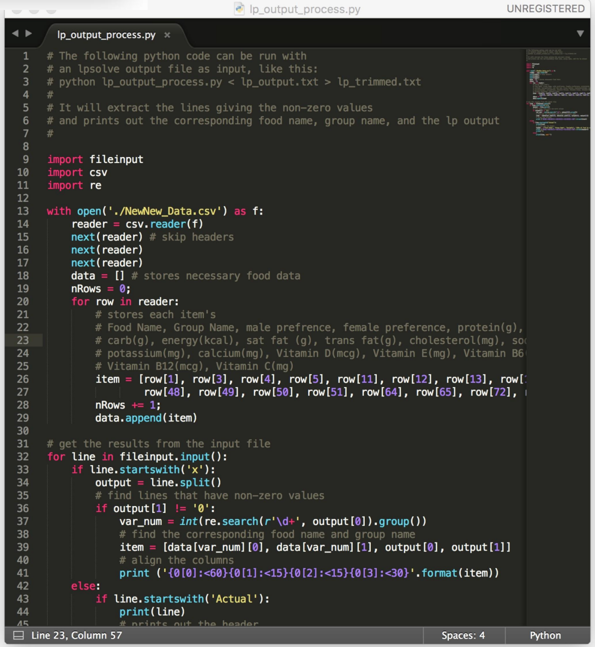
Implementation

1. Data Arrangement
2. Lpsolve Script Generation
3. Lpsolve Analysis

Script: Python
OS: Windows 10
SW: Eclipse, Python 3.5.1



```
Value of objective function: 1794.33880959

Actual values of the variables:
x_0                                 0
x_1                                 0
x_2                                 0
x_3                                 0
x_4                                 0
x_5                                 0
x_6                                 0
x_7                                 0
x_8                                 0
x_9                                 0
x_10                                0
x_11                                0
x_12                                0
x_13                                0
x_14                                0
x_15                                0
x_16                                0
x_17                                0
x_18                                0
x_19                                0
x_20                                0
x_21                                0
x_22                                0
x_23                                0
x_24                                0
x_25                                0
```



```python
# The following python code can be run with
# an lpsolve output file as input, like this:
# python lp_output_process.py < lp_output.txt > lp_trimmed.txt
#
# It will extract the lines giving the non-zero values
# and prints out the corresponding food name, group name, and the lp output
#

import fileinput
import csv
import re

with open('./NewNew_Data.csv') as f:
    reader = csv.reader(f)
    next(reader) # skip headers
    next(reader)
    next(reader)
    data = [] # stores necessary food data
    nRows = 0;
    for row in reader:
        # stores each item's
        # Food Name, Group Name, male prefrence, female preference, protein(g),
        # carb(g), energy(kcal), sat fat (g), trans fat(g), cholesterol(mg), so
        # potassium(mg), calcium(mg), Vitamin D(mcg), Vitamin E(mg), Vitamin B6
        # Vitamin B12(mcg), Vitamin C(mg)
        item = [row[1], row[3], row[4], row[5], row[11], row[12], row[13], row[
                row[48], row[49], row[50], row[51], row[64], row[65], row[72],
        nRows += 1;
        data.append(item)

# get the results from the input file
for line in fileinput.input():
    if line.startswith('x'):
        output = line.split()
        # find lines that have non-zero values
        if output[1] != '0':
            var_num = int(re.search(r'\d+', output[0]).group())
            # find the corresponding food name and group name
            item = [data[var_num][0], data[var_num][1], output[0], output[1]]
            # align the columns
            print ('{0[0]:<60}{0[1]:<15}{0[2]:<15}{0[3]:<30}'.format(item))
    else:
        if line.startswith('Actual'):
            print(line)
            # prints out the header
```

Solution: Male

Value of objective function: **1794.33880959**

Actual values of the variables:

Food Name	**Group Name**	**Variable**	**100g of food to consume daily (opt value)**
Chicory, pale variety, boiled in unsalted water	DG	x_716	30
Cranberries	FA	x_853	30
Duck, raw, meat only, weighed with fat, skin and bone	MCC	x_1049	2.94369
Fennel, Florence, boiled in unsalted water	DG	x_1089	8.29889
Fruit juice drink/squash, no sugar added, diluted	PCC	x_1162	30
Gourd, ridge, raw	DG	x_1199	21.6347
Jackfish, raw	JC	x_1316	0.46012
Jelly, sugar free, made with water	BR	x_1327	30
Lemon juice, fresh, weighed as whole fruit	FC	x_1492	30
Lime juice, fresh	FC	x_1520	15.6523
Marrow, boiled in unsalted water	DG	x_1575	30
Mushrooms, white, raw	DG	x_1699	3.7122
Orange juice, ambient, UHT	FC	x_1760	0.0189024
Passion fruit, flesh and pips, weighed with skin	FA	x_1817	30
Rock Salmon/Dogfish, raw	JA	x_2346	0.990351

Solution: Female

Value of objective function: **1639.81681676**

Actual values of the variables:

Food Name	**Group Name**	**Variable**	**100g of food to consume daily (opt value)**
Chicory, pale variety, boiled in unsalted water	DG	x_716	30
Cranberries	FA	x_853	30
Fennel, Florence, boiled in unsalted water	DG	x_1089	9.78434
Gourd, ridge, raw	DG	x_1199	2.39609
Jackfish, raw	JC	x_1316	0.809946
Jelly, sugar free, made with water	BR	x_1327	30
Lemon juice, fresh, weighed as whole fruit	FC	x_1492	30
Limes, flesh only, weighed with peel and pips	FA	x_1522	20.8073
Marrow, boiled in unsalted water	DG	x_1575	30
Mushrooms, white, raw	DG	x_1699	3.72073
Nutmeg, ground	H	x_1731	0.269869
Orange juice, ambient, UHT	FC	x_1760	0.0146341
Passion fruit, flesh and pips, weighed with skin	FA	x_1817	25.8045
Rock Salmon/Dogfish, raw	JA	x_2346	0.490755
Tea, black, infusion, average	PAA	x_2647	30

Variations

- Unhappy case: minimize our satisfaction but in a healthy way
    - Try to minimize our happiness and keep enough daily nutrition
    - the value of objective functions for male (8.33620465) and female (15.27183148) both decrease sharply
    - the numbers of food in the optimal food list for male and female are less than original model
    - Male: seven kinds of food (originally fifteen)
    - Female: nine kinds of food (originally fifteen)

Variations

- Vegetarian case: maximize our preference with a vegetarian diet
  - We modify our model and add some constraints that set the amount of all the food that made with meat and fish to zero
  - optimal happiness scores $\mathbf{H}_{max}$= 1454.78168382 for male,

    $\mathbf{H}_{max}$= 1296.64981001 for female
  - substituting meat and fish will reduce our happiness when we have the same nutrient constraints as before
  - consume more cereals and dairy to fulfill the nutrient need

Variations

- Diversity case: maximize our happiness with plenty of diversity of food items
  - We modify the original model to enlarge our possibility to getting more different types of food
  - we bound each group with an upper limit so that we cannot eat more than 1000 grams of any food for a single day
  - The maximum happiness score drop from 1794.33880959 to 618.16189370 for male and from 1639.81681676 to 558.85880275 for female
  - male can consume food from 14 kind of groups,
    female can consume food from 13 different groups,
    originally both male and female eat food from 8 different groups

Conclusions

From original study

- To maximize our **happiness**, a daily diet should consist of mostly fruits and vegetables.
- For non-vegetarians, fish seems to be the most nutritious meat type.
- Debatably, from our results, male seem to be more easily satisfied than female.

From variation studies

- To make oneself unhappy, breakfast cereal and milk/cheese might be good choices for male and female respectively.
- For vegetarians, with vegetables and fruits they could never be alone and sick.
- Having a diverse diet → a more balanced diet + a lower happiness level.

Conclusions

Limitations:

- Preference scores based on the small survey within our group (n=4), not representative
- The survey was inevitably focused on each food group, instead of individual food.
- Based on the dataset by Public Health England, cannot cover all U.S. foods

Further studies:

- Preference scores based on bigger survey randomly sampled from different populations.
- Survey focusing on individual food.
- Based on the dataset by U.S. Department of Agriculture
- Personal diet based on individuals

References

1. American Heart Association, “Know Your Facts” http://www.heart.org/HEARTORG/Conditions/Cholesterol/PreventionTreatmentofHighCholesterol/Know-Your-Fats_UCM_305628_Article.jsp
2. ConsumerLab, “Recommended Daily Intakes and Upper Limits for Nutrients” http://www.consumerlab.com/RDAs/
3. Sourceforge, “lpsolve” https://sourceforge.net/projects/lpsolve/
4. NEOS, “The Diet Problem” http://www.neos-guide.org/content/diet-problem
5. Public Health England, “McCance and Widdowson’s The Composition of Foods Integrated Dataset 2015” https://www.gov.uk/government/publications/composition-of-foods-integrated-dataset-cofid
6. SF Gate, “Daily Amounts of Carbs, Fat, Fiber, Sodium & Protein” http://healthyeating.sfgate.com/daily-amounts-carbs-fat-fiber-sodium-protein-4230.html
7. Sparkpeople, “Healthy Beverage Guidelines” http://www.sparkpeople.com/resource/nutrition_articles.asp?id=605
8. US Department of Argriculture, “Estimated Calorie Needs per Day by Age, Gender, and Physical Activity Level” https://www.cnpp.usda.gov/sites/default/files/usda_food_patterns/EstimatedCalorieNeedsPerDayTable.pdf
9. US Food and Drug Administration, “Guidance for Industry: A Food Labeling Guide (14. Appendix F: Calculate the Percent Daily Value for the Appropriate Nutrients)” http://www.fda.gov/Food/GuidanceRegulation/GuidanceDocumentsRegulatoryInformation/LabelingNutrition/ucm064928.htm
10. Vegetarian Resource Group, “Veganism in a Nutshell” http://www.vrg.org/nutshell/vegan.htm#what
11. Wikipedia, “Knapsack problem” https://en.wikipedia.org/wiki/Knapsack_problem
12. Wikipedia, “Stigler Diet” https://en.wikipedia.org/wiki/Stigler_diet

Questions?