# Project2_0

March 3, 2023

## 1 EEP 153 Project 2 Subsistence Cost Diet: Atwater

[1]:
```
%pip install -r requirements.txt
```

```
Requirement already satisfied: pint>=0.18 in /opt/conda/lib/python3.9/site-
packages (from -r requirements.txt (line 3)) (0.20.1)
Requirement already satisfied: requests>=2.26.0 in
/opt/conda/lib/python3.9/site-packages (from -r requirements.txt (line 6))
(2.26.0)
Requirement already satisfied: python-gnupg in /opt/conda/lib/python3.9/site-
packages (from -r requirements.txt (line 8)) (0.5.0)
Requirement already satisfied: eep153_tools in /opt/conda/lib/python3.9/site-
packages (from -r requirements.txt (line 10)) (0.11)
Requirement already satisfied: fooddatacentral in /opt/conda/lib/python3.9/site-
packages (from -r requirements.txt (line 12)) (1.0.9)
Requirement already satisfied: charset-normalizer~=2.0.0 in
/opt/conda/lib/python3.9/site-packages (from requests>=2.26.0->-r
requirements.txt (line 6)) (2.0.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/opt/conda/lib/python3.9/site-packages (from requests>=2.26.0->-r
requirements.txt (line 6)) (1.26.7)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.9/site-
packages (from requests>=2.26.0->-r requirements.txt (line 6)) (3.1)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/lib/python3.9/site-packages (from requests>=2.26.0->-r
requirements.txt (line 6)) (2021.10.8)
Note: you may need to restart the kernel to use updated packages.
```

[2]:
```
# API key for Gov;
apikey = "bwFohFv0W79JagEjhjfy121CHf29UEljz0OYel1N"
```

### 1.1 Dietary Reference Intakes

[3]:
```
# read in dietary requirements (max and min)
import pandas as pd
dri_max = pd.read_csv("Dietary Requirements Max.csv").set_index('Nutrition')
```

```python
dri_min = pd.read_csv("Dietary Requirements Min.csv").set_index('Nutrition')

# convert units from kcal to kJ
temp = dri_max.loc['Energy']
temp.iloc[1:] = temp.iloc[1:] * 4.184
dri_max.loc['Energy'] = temp

temp = dri_min.loc['Energy']
temp.iloc[1:] = temp.iloc[1:] * 4.184
dri_min.loc['Energy'] = temp
```

```
/opt/conda/lib/python3.9/site-packages/pandas/core/indexing.py:1965:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self.obj._check_is_chained_assignment_possible()
/opt/conda/lib/python3.9/site-packages/pandas/core/indexing.py:1732:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self._setitem_single_block(indexer, value, name)
```

[4]:

```python
import re
# function for age-sex specific DRI
def dri(age,sex,dietmin,dietmax):
    if age <= 3:
        index = 2

    else:
        for i, j in enumerate(list(dietmax)[3:]):
            if j[0] == sex:
                interval = re.findall(r'\d+', j)
                if len(interval) == 1:
                    if age >= int(interval[0]):
                        index = i + 3
                        break
                else:
                    if age >= int(interval[0]) and age <= int(interval[1]):
                        index = i + 3
                        break

    df = pd.DataFrame({'Nutrition': dietmin.iloc[:, 0],
                       'Max/Min': 'Minimum',
                       'Age & Sex': list(dietmin)[index],
                      'Intake': dietmin.iloc[:, index]})
    df_max = pd.DataFrame({'Nutrition': dietmax.iloc[:, 0],
                       'Max/Min': 'Maximum',
                       'Age & Sex': list(dietmax)[index],
                      'Intake': dietmax.iloc[:, index]})

    df = pd.concat([df, df_max], axis=0)

    return df
```

[5]:

```
# Testing: dietary reference intakes for 20-year-old female
dri(20, 'F', dri_min, dri_max)
```

[5]:

```
                               Nutrition  Max/Min Age & Sex   Intake
Nutrition
Energy                               ---  Minimum   F 19-30   8368.0
Protein                              RDA  Minimum   F 19-30     46.0
Fiber, total dietary                 ---  Minimum   F 19-30     28.0
Folate, DFE                          RDA  Minimum   F 19-30    400.0
Calcium, Ca                          RDA  Minimum   F 19-30   1000.0
Carbohydrate, by difference          RDA  Minimum   F 19-30    130.0
Iron, Fe                             RDA  Minimum   F 19-30     18.0
Magnesium, Mg                        RDA  Minimum   F 19-30    310.0
Niacin                               RDA  Minimum   F 19-30     14.0
Phosphorus, P                        RDA  Minimum   F 19-30    700.0
Potassium, K                          AI  Minimum   F 19-30   4700.0
Riboflavin                           RDA  Minimum   F 19-30      1.1
Thiamin                              RDA  Minimum   F 19-30      1.1
Vitamin A, RAE                       RDA  Minimum   F 19-30    700.0
Vitamin B-12                         RDA  Minimum   F 19-30      2.4
Vitamin B-6                          RDA  Minimum   F 19-30      1.3
Vitamin C, total ascorbic acid       RDA  Minimum   F 19-30     75.0
Vitamin E (alpha-tocopherol)         RDA  Minimum   F 19-30     15.0
Vitamin K (phylloquinone)             AI  Minimum   F 19-30     90.0
Zinc, Zn                             RDA  Minimum   F 19-30      8.0
Sodium, Na                            UL  Maximum   F 19-30   2300.0
Energy                               NaN  Maximum   F 19-30  12970.4
```

### 1.2 Data on prices for different foods

[6]:

```
# get food list from four restaurants
food_list_total = pd.read_csv("FoodList.csv")
food_list_total = food_list_total.astype({"FDC": str})
food_list_total
```

[6]:

```
     Restaurant                  Dish  Ingredients  Dish_Price  Quantity  \
0    Thai Basil  Pineapple Fried Rice       prawns       17.45  140.0000
1    Thai Basil  Pineapple Fried Rice      chicken       17.45  200.0000
2    Thai Basil  Pineapple Fried Rice          egg       17.45   48.0000
3    Thai Basil  Pineapple Fried Rice    Pineapple       17.45  180.0000
4    Thai Basil  Pineapple Fried Rice  white onion       17.45   80.0000
..          …                     …          …         …         …
181    Poke Bar      Wazzup Poke Bowl     cucumber       15.95   28.3500
182    Poke Bar      Wazzup Poke Bowl  green onion       15.95    7.0875
183    Poke Bar      Wazzup Poke Bowl        ponzu       15.95   35.4375
184    Poke Bar      Wazzup Poke Bowl       wasabi       15.95  452.1825
185    Poke Bar      Wazzup Poke Bowl         mayo       15.95   35.4375

     Unit      FDC  Calorie/100g  Ingredient_Price/100 gm
0    gram   175180          99.0                    1.200
1    gram   331960         152.5                    0.490
2    gram   748967         145.0                    0.500
3    gram  2346398          57.0                    0.520
4    gram  1104962          34.5                    0.498
..    …       …           …                    …
181  gram   168409          15.0                    0.880
182  gram   170006          27.0                    0.440
183  gram  2451144          33.0                    1.900
184  gram   171831         292.0                    8.780
185  gram   171002         334.0                    0.560

[186 rows x 9 columns]
```

[7]:

```python
# construct food lists by restaurant
grouped = food_list_total.groupby(food_list_total.Restaurant)
restaurants = ['Thai Basil', 'Ttoust', 'IB', 'Poke Bar']

food_list_res = []

for r in restaurants:
    food_list_res.append(grouped.get_group(r))
```

[8]:

```python
import fooddatacentral as fdc
import warnings
from collections import defaultdict

# get nutritional information for ingredients
ing_res = []
ing_res_list = []
for i in range(4):
    L = []
    D = {}
    items = []
    count = 0
    food_list = food_list_res[i]
    for food in food_list.Ingredients.tolist():
        try:
            FDC = food_list.iloc[count,:].FDC
            count+=1
            temp = fdc.nutrients(apikey,FDC)
            key = temp.Units
            # convert units if necessary
            if 'Energy' in key.index:
                if key['Energy'] != 'kJ':
                    temp.Quantity['Energy'] = temp.Quantity['Energy']*4.184

            L.append(temp.Quantity)
            D[food] = temp.Quantity
            items.append(food)
        except AttributeError:
            warnings.warn("Couldn't find FDC Code %s for food %s." % (food,FDC))

    # construct nutrient tables
    # list version
    FoodNutrients_Ing = pd.DataFrame(L,dtype=float)
    FoodNutrients_Ing.index = items
    FoodNutrients_Ing = FoodNutrients_Ing.fillna(0)
    FoodNutrients_Ing = FoodNutrients_Ing.transpose()
    ing_res_list.append(FoodNutrients_Ing)
    # dictionary version
    FoodNutrients_Ing_d = pd.DataFrame(D,dtype=float)
    FoodNutrients_Ing_d = FoodNutrients_Ing_d.fillna(0)
    ing_res.append(FoodNutrients_Ing_d)
```

```
/tmp/ipykernel_80/700468894.py:23: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  temp.Quantity['Energy'] = temp.Quantity['Energy']*4.184
```

[9]:

```
# example: nutrient table for Thai Basil
ing_res_list[0]
```

[9]:

```
                           prawns  chicken     egg  Pineapple  white onion  \
Proximates                   0.00     0.00    0.00   0.000000         0.00
Water                       74.33    65.30   75.80  84.990000        91.30
Energy                     415.00   695.00  617.00   0.000000       148.00
```

```
Protein                     23.98    32.10   12.40   0.460938         0.89
Total lipid (fat)            0.28     3.24    9.96   0.211300         0.13
…                           …      …     …        …          …
Beta-sitostanol              0.00     0.00    0.00   0.000000         0.00
Delta-5-avenasterol          0.00     0.00    0.00   0.000000         0.00
Delta-7-Stigmastenol         0.00     0.00    0.00   0.000000         0.00
Ergothioneine                0.00     0.00    0.00   0.000000         0.00
Vitamin K (Menaquinone-4)    0.00     0.00    0.00   0.000000         0.00

                            raisins  cashew nuts    peas   carrots  chicken  \
Proximates                    0.000         0.00    0.00   0.00000      0.0
Water                        15.500         5.20   78.86  87.72000     93.9
Energy                     1251.016      2314.00  339.00   0.00000     84.0
Protein                       3.300        18.22    5.42   0.94125      0.7
Total lipid (fat)             0.250        43.85    0.40   0.35060      0.1
…                             …          …     …       …      …
Beta-sitostanol               0.000         0.00    0.00   0.00000      0.0
Delta-5-avenasterol           0.000         0.00    0.00   0.00000      0.0
Delta-7-Stigmastenol          0.000         0.00    0.00   0.00000      0.0
Ergothioneine                 0.000         0.00    0.00   0.00000      0.0
Vitamin K (Menaquinone-4)     0.000         0.00    0.00   0.00000      0.0

                           …      pork  imitation crab  prawns  mushrooms  \
Proximates                 …      0.00            0.00    0.00   0.000000
Water                      …     24.76           74.70   74.33  88.600000
Energy                     …  2645.00          397.48  415.00   0.000000
Protein                    …      9.25            7.62   23.98   2.414375
Total lipid (fat)          …     65.70            0.46    0.28   0.195000
…                        …      …             …     …        …
Beta-sitostanol            …      0.00            0.00    0.00   0.000000
Delta-5-avenasterol        …      0.00            0.00    0.00   0.000000
Delta-7-Stigmastenol       …      0.00            0.00    0.00   0.000000
Ergothioneine              …      0.00            0.00    0.00  11.060000
Vitamin K (Menaquinone-4)  …      0.00            0.00    0.00   0.000000

                           white onion     rice  catfish  eggplant   \
Proximates                        0.00     0.00     0.00       0.00
Water                            91.30    11.60    79.06      92.30
Energy                          148.00  1500.00   496.00     104.00
Protein                           0.89     6.94    15.23       0.98
Total lipid (fat)                 0.13     1.30     5.94       0.18
…                                …      …      …        …
Beta-sitostanol                   0.00     0.00     0.00       0.00
Delta-5-avenasterol               0.00     0.00     0.00       0.00
Delta-7-Stigmastenol              0.00     0.00     0.00       0.00
Ergothioneine                     0.00     0.00     0.00       0.00
Vitamin K (Menaquinone-4)         0.00     0.00     0.60       0.00
```

```
                           red curry     rice
Proximates                     0.000     0.00
Water                          0.000    11.60
Energy                       556.472  1500.00
Protein                        6.670     6.94
Total lipid (fat)              3.330     1.30
…                                …      …
Beta-sitostanol                0.000     0.00
Delta-5-avenasterol            0.000     0.00
Delta-7-Stigmastenol           0.000     0.00
Ergothioneine                  0.000     0.00
Vitamin K (Menaquinone-4)      0.000     0.00

[206 rows x 41 columns]
```

[10]:

```
# construct price vector for dishes
# Convert food quantities to FDC units
food_list_total['FDC Quantity'] = food_list_total[['Quantity','Unit']].T.
 ↪apply(lambda x : fdc.units(x['Quantity'],x['Unit']))

food_list_total['FDC Price'] = food_list_total['Dish_Price']/food_list_total.
 ↪groupby('Dish',sort=False)['FDC Quantity'].sum()

food_list_total.dropna(how='any')

Dish_Prices = food_list_total.groupby('Dish',sort=False)['Dish_Price'].min()/
 ↪food_list_total.groupby('Dish',sort=False)['FDC Quantity'].sum()
```

```
/opt/conda/lib/python3.9/site-packages/pandas/core/dtypes/cast.py:1990:
UnitStrippedWarning: The unit of the quantity is stripped when downcasting to
ndarray.
  result[:] = values
```

### 1.3 Table of nutritional information for dishes

[11]:

```
# construct nutrient table for dishes
FoodNutrients_Ing_total = pd.DataFrame()
for t in ing_res_list:
    FoodNutrients_Ing_total = pd.concat([FoodNutrients_Ing_total, t], axis=1)

FoodNutrients_Ing_total = FoodNutrients_Ing_total.fillna(0)


dishes = food_list_total.groupby('Dish',sort=False)['Ingredients'].count()

FoodNutrients = pd.DataFrame(columns=dishes.keys().tolist(),
                    index=FoodNutrients_Ing.index, )

# sum up nutrients of each ingredient of the dishes
start = 0
for i, column in enumerate(FoodNutrients):

    FoodNutrients[column] = FoodNutrients_Ing_total.iloc[:, start:
 ↪start+dishes[i]].sum(axis=1)
    start = start+dishes[i]

FoodNutrients
```

[11]:

```
                                    Pineapple Fried Rice  \
Proximates                                      0.000000
Water                                         579.000000
Energy                                       5779.016000
Protein                                        97.712187
Total lipid (fat)                              58.671900
…                                                    …
Fatty acids, total trans-polyenoic              0.000000
Fluoride, F                                     0.000000
Phytosterols                                    0.000000
Sugars, added                                   0.000000
PUFA 18:3i                                      0.000000

                                    Basil Eggplant Chicken  \
Proximates                                        0.000000
Water                                           473.040000
Energy                                         1930.000000
Protein                                          13.541875
Total lipid (fat)                                 2.505600
…                                                      …
Fatty acids, total trans-polyenoic                0.000000
Fluoride, F                                       0.000000
Phytosterols                                      7.000000
Sugars, added                                     0.000000
PUFA 18:3i                                        0.000000

                                    Pad Kee Mow Chicken  \
Proximates                                     0.000000
Water                                        513.500000
Energy                                      2170.728000
Protein                                       25.511875
Total lipid (fat)                             11.285600
…                                                   …
Fatty acids, total trans-polyenoic             0.000000
Fluoride, F                                    0.000000
```

```
Phytosterols                                0.000000
Sugars, added                               0.000000
PUFA 18:3i                                  0.000000

                                    Pad Thai Chicken & Prawns  \
Proximates                                               0.00
Water                                                  474.33
Energy                                                4972.00
Protein                                                102.38
Total lipid (fat)                                       69.83
…                                                       …
Fatty acids, total trans-polyenoic                       0.00
Fluoride, F                                              0.00
Phytosterols                                             0.00
Sugars, added                                            0.00
PUFA 18:3i                                               0.00

                                    Combo Garlic Pepper  Spicy Basil Catfish  \
Proximates                                     0.000000                0.000
Water                                        504.290000              182.960
Energy                                      6312.480000             2656.472
Protein                                      106.594375               29.820
Total lipid (fat)                             73.785000               10.750
…                                                   …                    …
Fatty acids, total trans-polyenoic             0.132000                0.024
Fluoride, F                                    0.000000                0.000
Phytosterols                                   0.000000                7.000
Sugars, added                                  0.000000                0.000
PUFA 18:3i                                     0.000000                0.000

                                        Bibim Bop    Kimchi Soup  \
Proximates                               0.000000       0.000000
Water                                  514.180000     450.390000
Energy                              290094.731627  331088.272409
Protein                                 46.090000      97.120000
Total lipid (fat)                       14.040000     113.200000
…                                           …             …
Fatty acids, total trans-polyenoic       0.000000       0.132000
Fluoride, F                             41.000000       1.100000
Phytosterols                             0.000000      15.000000
Sugars, added                            0.000000       0.000000
PUFA 18:3i                               0.000000       0.000000

                                    Galbi Short Rib  Spicy Rice Cake  …  \
Proximates                                 0.000000         0.000000  …
Water                                    407.910000       542.520000  …
Energy                                363957.609597    326442.412815  …
```

```
Protein                                  115.390000        32.420000  …
Total lipid (fat)                         95.080000         5.040000  …
…                                              …              …   …
Fatty acids, total trans-polyenoic         0.000000         0.000000  …
Fluoride, F                                1.100000         6.400000  …
Phytosterols                             767.000000        41.000000  …
Sugars, added                            100.000000       100.000000  …
PUFA 18:3i                                 0.000000         0.000000  …

                                    Veggie Combo  IB's Original(chicken)  \
Proximates                              0.000000                0.000000   
Water                                 599.720000              497.990000   
Energy                             129317.731131           129265.731131   
Protein                                50.160000               79.820000   
Total lipid (fat)                     145.610000              133.010000   
…                                            …                       …   
Fatty acids, total trans-polyenoic      0.334000                0.334000   
Fluoride, F                             2.300000                2.300000   
Phytosterols                           45.000000               45.000000   
Sugars, added                           0.000000                0.000000   
PUFA 18:3i                              0.003000                0.003000   

                                    IB's Original(beef)  IB's Original(lamb)  \
Proximates                                     0.000000             0.000000   
Water                                        506.390000           492.160000   
Energy                                    129082.731131        129750.731131   
Protein                                       71.120000            64.280000   
Total lipid (fat)                            132.250000           153.180000   
…                                                   …                    …   
Fatty acids, total trans-polyenoic             0.334000             0.334000   
Fluoride, F                                    2.300000             2.300000   
Phytosterols                                  45.000000            45.000000   
Sugars, added                                  0.000000             0.000000   
PUFA 18:3i                                     0.003000             0.003000   

                                    Original Salmon Poke Bowl  \
Proximates                                           0.000000   
Water                                              349.970000   
Energy                                           14278.841536   
Protein                                             42.170000   
Total lipid (fat)                                   81.880000   
…                                                        …   
Fatty acids, total trans-polyenoic                   0.000000   
Fluoride, F                                          0.000000   
Phytosterols                                         0.000000   
Sugars, added                                        0.000000   
PUFA 18:3i                                           0.000000   
```

```
                                    Firecracker Poke Bowl  \
Proximates                                       0.000000
Water                                          349.370000
Energy                                       50999.296902
Protein                                         30.610000
Total lipid (fat)                              120.000000
…                                                   …
Fatty acids, total trans-polyenoic               0.000000
Fluoride, F                                      1.300000
Phytosterols                                    14.000000
Sugars, added                                   13.300000
PUFA 18:3i                                       0.000000

                                    Sunset House Poke Bowl  \
Proximates                                        0.000000
Water                                           424.540000
Energy                                        42197.319302
Protein                                          56.240000
Total lipid (fat)                                71.250000
…                                                     …
Fatty acids, total trans-polyenoic                0.000000
Fluoride, F                                       1.300000
Phytosterols                                     14.000000
Sugars, added                                    13.300000
PUFA 18:3i                                        0.000000

                                    The O.G. Poke Bowl  Goodie Mob Poke Bowl  \
Proximates                                     0.00000              0.000000
Water                                        336.83000            444.350000
Energy                                     44274.83239           5629.147776
Protein                                       44.55000             41.900000
Total lipid (fat)                             63.89000             13.850000
…                                                 …                   …
Fatty acids, total trans-polyenoic             0.00000              0.000000
Fluoride, F                                    0.00000              1.300000
Phytosterols                                   0.00000             14.000000
Sugars, added                                 13.30000              0.000000
PUFA 18:3i                                     0.00000              0.000000

                                    Wazzup Poke Bowl
Proximates                                  0.000000
Water                                     448.360000
Energy                                   5466.693248
Protein                                    33.200000
Total lipid (fat)                          46.890000
…                                               …
```

```
Fatty acids, total trans-polyenoic          0.023000
Fluoride, F                                 1.300000
Phytosterols                               14.000000
Sugars, added                               0.000000
PUFA 18:3i                                  0.000000

[144 rows x 23 columns]
```

[12]:

```
# price vector for dishes
Dish_Prices
```

[12]:

```
Dish
Pineapple Fried Rice          2.326666666666666 / hectogram
Basil Eggplant Chicken       1.9074074074074074 / hectogram
Pad Kee Mow Chicken           2.653225806451613 / hectogram
Pad Thai Chicken & Prawns     1.852497096399535 / hectogram
Combo Garlic Pepper          1.8911335578002246 / hectogram
Spicy Basil Catfish          1.8994708994708995 / hectogram
Bibim Bop                     3.021598272138229 / hectogram
Kimchi Soup                  2.7539370078740157 / hectogram
Galbi Short Rib               4.401988636363637 / hectogram
Spicy Rice Cake               1.635846372688478 / hectogram
BBQ Chicken                   2.013888888888889 / hectogram
Cheeseburger                  2.472527472527472 / hectogram
Avogobble sandwich           3.1055900621118013 / hectogram
Veggie Combo                  3.535353535353536 / hectogram
IB's Original(chicken)       2.2569444444444446 / hectogram
IB's Original(beef)          2.2569444444444446 / hectogram
IB's Original(lamb)          2.2569444444444446 / hectogram
Original Salmon Poke Bowl    3.2149155958679763 / hectogram
Firecracker Poke Bowl         6.082272748939414 / hectogram
Sunset House Poke Bowl       2.6790963298899806 / hectogram
The O.G. Poke Bowl           1.7337757450751807 / hectogram
Goodie Mob Poke Bowl         1.7472367368847699 / hectogram
Wazzup Poke Bowl             1.6819438842358623 / hectogram
dtype: object
```

[13]:

```
# construct price vector for ingredients
ing_prices = []
for r in food_list_res:
    food_list = r
    food_list['FDC Quantity'] = food_list[['Quantity','Unit']].T.apply(lambda x␣
 ↪: fdc.units(x['Quantity'],x['Unit']))

    # Now may want to filter df by time or place--need to get a unique set of␣
 ↪food names.
    food_list['FDC Price'] = food_list['Ingredient_Price/100 gm']
    food_list.dropna(how='any') # Drop food with any missing data

    # To use minimum price observed
    Ingredient_Prices = food_list.groupby('Ingredients',sort=False)['FDC␣
↪Price'].min()
    ing_prices.append(Ingredient_Prices)
```

```
/opt/conda/lib/python3.9/site-packages/pandas/core/dtypes/cast.py:1990:
UnitStrippedWarning: The unit of the quantity is stripped when downcasting to
ndarray.
  result[:] = values
/tmp/ipykernel_80/1380384181.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  food_list['FDC Quantity'] = food_list[['Quantity','Unit']].T.apply(lambda x :
fdc.units(x['Quantity'],x['Unit']))
/tmp/ipykernel_80/1380384181.py:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  food_list['FDC Price'] = food_list['Ingredient_Price/100 gm']
```

[14]:
```
# Example: ingredient prices for Poke bar (per 100g)
ing_prices[3]
```

[14]:
```
Ingredients
salmon            2.20
white rice        0.28
sweet onion       0.44
green onion       0.44
kale              0.88
shoyu             1.69
spicy mayo        1.38
ahi tuna          2.36
cucumber          0.88
sweet chili       1.05
house dressing    3.17
edamame           0.56
ponzu             1.90
tofu              0.74
wasabi            8.78
```

```
mayo              0.56
Name: FDC Price, dtype: float64
```

### 1.4 Solution

```
[15]: from  scipy.optimize import linprog as lp
      import numpy as np
      import warnings

      def␣
       ↪solve_subsistence_problem(FoodNutrients,Prices,dietmin,dietmax,max_weight=None,tol=1e-6):
       ↪
          """Solve Stigler's Subsistence Cost Problem.

          Inputs:
             - FoodNutrients : A pd.DataFrame with rows corresponding to foods,␣
       ↪columns to nutrients.
             - Prices : A pd.Series of prices for different foods
             - diet_min : A pd.Series of DRIs, with index corresponding to columns of␣
       ↪FoodNutrients,
                          describing minimum intakes.
             - diet_max : A pd.Series of DRIs, with index corresponding to columns of␣
       ↪FoodNutrients,
                          describing maximum intakes.
             - max_weight : Maximum weight (in hectograms) allowed for diet.
             - tol : Solution values smaller than this in absolute value treated as␣
       ↪zeros.

          """
          try:
              p = Prices.apply(lambda x:x.magnitude)
          except AttributeError:  # Maybe not passing in prices with units?
              warnings.warn("Prices have no units.  BE CAREFUL!  We're assuming␣
       ↪prices are per hectogram or deciliter!")
              p = Prices

          p = p.dropna()

          # Compile list that we have both prices and nutritional info for; drop if␣
       ↪either missing
          use = p.index.intersection(FoodNutrients.columns)
          p = p[use]

          # Drop nutritional information for foods we don't know the price of,
          # and replace missing nutrients with zeros.
          Aall = FoodNutrients[p.index].fillna(0)
```

```
#print(Aall)
# Drop rows of A that we don't have constraints for.
Amin = Aall.loc[Aall.index.intersection(dietmin.index)]
#print(dietmin)
#print(Amin)
Amin = Amin.reindex(dietmin.index,axis=0)
#print(dietmin.index)
#print(Amin)
idx = Amin.index.to_frame()
#print(Amin)
idx['type'] = 'min'
#print(Amin)
#Amin.index = pd.MultiIndex.from_frame(idx)
#dietmin.index = Amin.index



Amax = Aall.loc[Aall.index.intersection(dietmax.index)]
Amax = Amax.reindex(dietmax.index,axis=0)
idx = Amax.index.to_frame()
idx['type'] = 'max'
#Amax.index = pd.MultiIndex.from_frame(idx)
#dietmax.index = Amax.index

# Minimum requirements involve multiplying constraint by -1 to make <=.
A = pd.concat([Amin,
               -Amax])

b = pd.concat([dietmin,
               -dietmax]) # Note sign change for max constraints

# Make sure order of p, A, b are consistent
A = A.reindex(p.index,axis=1)
A = A.reindex(b.index,axis=0)

if max_weight is not None:
    # Add up weights of foods consumed
    A.loc['Hectograms'] = -1
    b.loc['Hectograms'] = -max_weight
#print(p)
#print(A)
#print(b)
# Now solve problem!  (Note that the linear program solver we'll use assumes
# "less-than-or-equal" constraints.  We can switch back and forth by
# multiplying $A$ and $b$ by $-1$.)

result = lp(p, -A, -b, method='interior-point', options={'presolve': True,
                                                        'cholesky':False,
                                                        'sym_pos':False,
                                                        'lstsq':True})

result.A = A
result.b = b

if result.success:
    result.diet = pd.Series(result.x,index=p.index)
else: # No feasible solution?
    warnings.warn(result.message)
    result.diet = pd.Series(result.x,index=p.index)*np.nan

return result
```

### 1.4.1 Dish-based Solution

[16]:

```
#dish-based result
group = 'F 19-30'
tol = 1e-6

result =␣
  ↪solve_subsistence_problem(FoodNutrients,Dish_Prices,dri_min[group],dri_max[group],tol=tol)

print("Cost of diet for %s is $%4.2f per day.\n" % (group,result.fun))

# Put back into nice series
diet = result.diet

print("\nDiet (in 100s of grams or milliliters):")
print(diet[diet >= tol])  # Drop items with quantities less than precision of␣
  ↪calculation.
print()

tab = pd.DataFrame({"Outcome":np.abs(result.A).dot(diet),"Recommendation":np.
  ↪abs(result.b)})
print("\nWith the following nutritional outcomes of interest:")
print(tab)
print()

print("\nConstraining nutrients are:")
excess = tab.diff(axis=1).iloc[:,1]
print(excess.loc[np.abs(excess) < tol*100].index.tolist())
```

```
Cost of diet for F 19-30 is $5.61 per day.
```

```
Diet (in 100s of grams or milliliters):
Pineapple Fried Rice         0.241899
Basil Eggplant Chicken       0.500536
Pad Thai Chicken & Prawns    1.971385
Cheeseburger                 0.176456
dtype: float64


With the following nutritional outcomes of interest:
                                    Outcome  Recommendation
Nutrition
Energy                         12970.400054          8368.0
Protein                          244.298778            46.0
Fiber, total dietary              30.592201            28.0
Folate, DFE                      693.300183           400.0
Calcium, Ca                     1144.830377          1000.0
Carbohydrate, by difference      198.625688           130.0
Iron, Fe                          23.175260            18.0
Magnesium, Mg                    729.030397           310.0
Niacin                            48.393207            14.0
Phosphorus, P                   2715.994864           700.0
Potassium, K                    4700.000002          4700.0
Riboflavin                         2.504842             1.1
Thiamin                            2.841654             1.1
Vitamin A, RAE                   700.000008           700.0
Vitamin B-12                       3.858840             2.4
Vitamin B-6                        2.827509             1.3
Vitamin C, total ascorbic acid   232.956043            75.0
Vitamin E (alpha-tocopherol)      15.000000            15.0
Vitamin K (phylloquinone)        253.555642            90.0
Zinc, Zn                          26.728344             8.0
Sodium, Na                      1580.312476          2300.0
Energy                         12970.400054         12970.4


Constraining nutrients are:
['Potassium, K', 'Vitamin A, RAE', 'Vitamin E (alpha-tocopherol)', 'Energy']
```

### 1.4.2 Ingredient-based solutions

[18]:
```
# ingredient-based results for each of 4 restaurants
groups = dri_min.columns[1:]

cost_tbl = pd.DataFrame(columns = groups, index = restaurants)
diet_tbl_name = pd.DataFrame(columns = groups, index = restaurants)
diet_tbl = pd.DataFrame(columns = groups, index = restaurants)
nurt_tbl = pd.DataFrame(columns = groups, index = restaurants)
```

```python
excess_tbl = pd.DataFrame(columns = groups, index = restaurants)

for group in groups:
    tol = 1e-6
    for i in range(4):
        result = solve_subsistence_problem(ing_res[i],ing_prices[i],dri_min[group],dri_max[group],tol=tol)

        #results_tbl.loc[restaurants[i], group] = result

        #print("Result for restaurant: " + restaurants[i])
        #print("Cost of diet for %s is $%4.2f per day.\n" % (group,result.fun))

        cost_tbl.loc[restaurants[i], group] = result.fun

        # Put back into nice series
        diet = result.diet

        #print("\nDiet (in 100s of grams or milliliters):")
        #print(diet[diet >= tol])  # Drop items with quantities less than precision of calculation.
        #print()

        diet_tbl_name.loc[restaurants[i], group] = list(diet[diet >= tol].index)
        diet_tbl.loc[restaurants[i], group] = list(diet[diet >= tol])

        tab = pd.DataFrame({"Outcome":np.abs(result.A).dot(diet),"Recommendation":np.abs(result.b)})
        #print("\nWith the following nutritional outcomes of interest:")
        #print(tab)
        #print()

        nurt_tbl.loc[restaurants[i], group] = list(np.abs(result.A).dot(diet))

        #print("\nConstraining nutrients are:")
        excess = tab.diff(axis=1).iloc[:,1]
        #print(excess.loc[np.abs(excess) < tol*100].index.tolist())

        excess_tbl.loc[restaurants[i], group] = list(excess.loc[np.abs(excess) < tol*100].index.tolist())
```

```
/tmp/ipykernel_80/2046845037.py:22: UserWarning: Prices have no units.  BE
CAREFUL!  We're assuming prices are per hectogram or deciliter!
  warnings.warn("Prices have no units.  BE CAREFUL!  We're assuming prices are
per hectogram or deciliter!")
```

[19]:
```
# table of minimum diets costs by age-sex groups and restaurants
cost_tbl
```

[19]:
```
                C 1-3     F 4-8     M 4-8     F 9-13     M 9-13    F 14-18  \
Thai Basil   4.052806  5.681346  5.681348   7.366037   7.347198   7.559409
Ttoust       3.254694  4.524609  4.524609   5.881514    5.88149   6.141627
IB           5.046551  5.973081   5.97308   6.050696   6.050574   9.070692
Poke Bar     6.377665  8.560379  8.560379  10.494406  10.194235  12.590013

               M 14-18    F 19-30    M 19-30    F 31-50    M 31-50     F 51+  \
Thai Basil    7.889342   6.257664   6.693508   6.257664   6.693506   7.163111
Ttoust        6.160926   5.353081    5.41441   5.353081    5.41441   5.832565
IB            7.207749  10.594395   7.080005  10.594395   7.198854   6.485626
Poke Bar     15.451584  12.504119  15.451584  12.504119  15.451586  12.504119

                 M 51+
Thai Basil    6.693506
Ttoust         5.41441
IB            7.198853
Poke Bar     15.451584
```

[20]:
```
# table of minimum cost diet compositions by age-sex groups and restaurants
diet_tbl
# it seems there is a feasible solution for each group-restaurant combination
```

[20]:
```
                                                          C 1-3  \
Thai Basil   [1.6535947706889553, 1.3852752287717585, 0.709…
Ttoust       [0.882352925603457, 1.5314822731509792, 0.2748…
IB           [0.12140266060111793, 0.8973065484934414, 2.49…
Poke Bar     [0.2786377700974017, 1.04206165823365, 4.13955…

                                                          F 4-8  \
Thai Basil   [2.1982570758309756, 4.11231494774835, 0.34127…
Ttoust       [1.1764705868476217, 2.551020416219487, 0.4147…
IB           [0.7381884201246615, 0.36265548456224805, 0.60…
Poke Bar     [0.3715170277047409, 0.923279850987226, 6.4896…

                                                          M 4-8  \
Thai Basil   [2.198256851418826, 1.78151173378974e-06, 4.11…
Ttoust       [1.1764705880371733, 2.5510204034365738, 0.414…
IB           [0.7381884087079783, 0.3626555521447387, 0.607…
Poke Bar     [0.37151702781763934, 0.9232798518582935, 6.48…

                                                         F 9-13  \
Thai Basil   [3.3071895265853226, 5.56034958804642, 0.07673…
Ttoust       [1.7647058822237875, 4.591836734301037, 0.4271…
IB           [0.1818589658506171, 0.9773167656717172, 0.555…
```

```
Poke Bar    [0.6315231559302386, 0.8115626100037986, 1.611…

                                                      M 9-13  \
Thai Basil  [3.3071895423056197, 4.722821936295839, 0.1153…
Ttoust      [1.7647058820567403, 4.591836733460813, 0.4222…
IB          [0.18185436510819952, 0.9757250262422656, 0.55…
Poke Bar    [0.593710832494077, 0.7905590873998726, 1.4035…

                                                     F 14-18  \
Thai Basil  [3.8562091513207064, 4.790331164253986, 0.0718…
Ttoust      [2.3529411759875267, 6.632653048006388, 0.1522…
IB          [1.2233763054814664, 1.5757633454140303e-06, 2…
Poke Bar    [1.8430382611738536, 0.8342380454859687, 7.086…

                                                     M 14-18  \
Thai Basil  [4.925740434841197, 0.20758255836004372, 4.847…
Ttoust      [2.352941175766801, 0.038373509910166095, 7.42…
IB          [0.500016788304174, 0.0250903584780613, 1.0265…
Poke Bar    [0.6937842576074815, 0.48377537465033155, 12.9…

                                                     F 19-30  \
Thai Basil  [3.849673202078504, 3.9686114560475874, 0.1365…
Ttoust      [0.899093957300926, 0.2563041354425244, 6.1623…
IB          [1.5156130817038338, 3.8785864810287785, 0.173…
Poke Bar    [1.8802723382680828, 0.8456054989862727, 6.898…

                                                     M 19-30  \
Thai Basil  [4.947712189157788, 1.9233930755712387e-06, 3…
Ttoust      [0.7481425468840616, 0.515229448839125, 6.1869…
IB          [0.5156678917103537, 0.3145205737629059, 1.203…
Poke Bar    [0.6937842973729921, 0.4837754199649512, 12.93…

                                                     F 31-50  \
Thai Basil  [3.849673201126317, 3.968611957532809, 0.13658…
Ttoust      [0.8990940417231688, 0.2563041491157663, 6.162…
IB          [1.5156130812065853, 3.8785864897327436, 0.173…
Poke Bar    [1.8802723187799568, 0.8456055203133529, 6.898…

                                                     M 31-50  \
Thai Basil  [4.94771241775595, 3.8465795947305943, 0.17998…
Ttoust      [0.7481426971027444, 0.5152294766063318, 6.186…
IB          [0.41682131618855045, 0.22160690879455405, 1.0…
Poke Bar    [0.6937844327917748, 0.4837758079467785, 2.984…

                                                       F 51+  \
Thai Basil  [3.8496732024200333, 4.427298237925396, 0.0987…
Ttoust      [2.3529411763500763, 0.05379011503111703, 7.34…
```

```
IB          [0.618548819765312, 0.7183637701965844, 0.6563…
Poke Bar    [1.880272337983991, 0.8456054990230503, 6.8981…

                                                     M 51+
Thai Basil  [4.947712418235566, 3.8465794220684857, 0.1799…
Ttoust      [0.7481427032552965, 0.5152294773758332, 6.186…
IB          [0.4168214000171779, 0.22160785282771595, 1.01…
Poke Bar    [0.693784262838222, 0.4837754128303332, 12.933…
```

[21]:
```
# table of minimum cost diet nutrients by age-sex groups and restaurants
# saved for reference
nurt_tbl
```

[21]:
```
                                                     C 1-3  \
Thai Basil  [10459.999995221711, 129.5019976253498, 34.722…
Ttoust      [10459.953723745351, 70.15028613355202, 74.786…
IB          [10459.998723757919, 41.30292080187364, 64.805…
Poke Bar    [10459.999993487745, 61.559446143590854, 35.81…

                                                     F 4-8  \
Thai Basil  [10459.999798933926, 148.0204034319321, 43.729…
Ttoust      [10459.998923079153, 84.68608960743502, 86.219…
IB          [10459.999940213434, 66.02795640885225, 70.311…
Poke Bar    [10459.999992769399, 68.45313272332545, 44.659…

                                                     M 4-8  \
Thai Basil  [10459.997362561839, 148.02040123585098, 43.72…
Ttoust      [10459.999978625312, 84.68608968569447, 86.219…
IB          [10459.999994271622, 66.02795565673674, 70.311…
Poke Bar    [10459.999999013782, 68.45313276573496, 44.659…

                                                    F 9-13  \
Thai Basil  [11715.19977423415, 180.38533573952296, 51.190…
Ttoust      [11715.199995352059, 90.1035842617758, 80.1428…
IB          [11715.199936077588, 64.59512699468041, 70.985…
Poke Bar    [11715.199865684588, 82.17553711592485, 51.465…

                                                    M 9-13  \
Thai Basil  [12551.999993604923, 187.97605590671975, 51.43…
Ttoust      [12551.999976754489, 90.07261058475581, 80.161…
IB          [12551.998074774834, 64.59840709075098, 70.971…
Poke Bar    [12551.99970222906, 83.90854277631367, 51.7795…

                                                   F 14-18  \
Thai Basil  [12970.399993055571, 194.18686472777378, 52.54…
Ttoust      [12970.399980964057, 81.51082052138939, 58.522…
IB          [12970.39988761615, 89.80200217976017, 85.7913…
```

```
Poke Bar    [12970.399999938589, 105.70595888960699, 49.49…

                                                       M 14-18  \
Thai Basil  [12970.399999828469, 199.21876861795843, 51.22…
Ttoust      [12970.399995082578, 74.06760502559081, 48.198…
IB          [11713.816581729156, 73.49162617538681, 81.328…
Poke Bar    [12970.399999852943, 102.05234754739155, 73.92…

                                                       F 19-30  \
Thai Basil  [12970.399992646822, 178.40949632516484, 50.77…
Ttoust      [10247.893539693945, 72.49513357038175, 45.596…
IB          [12970.399998797377, 86.9498591895119, 91.5198…
Poke Bar    [12970.39999791503, 105.8802999677998, 48.7080…

                                                       M 19-30  \
Thai Basil  [12970.398166111434, 182.86940614655921, 48.40…
Ttoust      [10336.15056869431, 69.99744635487859, 41.8438…
IB          [11356.763389280424, 70.45991284831139, 78.935…
Poke Bar    [12970.399995953387, 102.05234729196603, 73.92…

                                                       F 31-50  \
Thai Basil  [12970.399946056566, 178.40949611303526, 50.77…
Ttoust      [10247.87581643742, 72.49513338072387, 45.5962…
IB          [12970.399997146023, 86.94985901816491, 91.519…
Poke Bar    [12970.399629448391, 105.88029803252115, 48.70…

                                                       M 31-50  \
Thai Basil  [12970.399993736613, 182.86941525890563, 48.40…
Ttoust      [10336.14318441441, 69.9974462732341, 41.84380…
IB          [12046.084109374315, 75.22743889530969, 83.184…
Poke Bar    [12970.399910000573, 102.05234644639543, 73.92…

                                                         F 51+  \
Thai Basil  [12970.399996065426, 188.83539023249273, 51.85…
Ttoust      [6694.400002435699, 74.36675287352797, 49.5088…
IB          [10656.800277004537, 69.9291372665744, 68.3051…
Poke Bar    [12970.399999125044, 105.88029996739327, 48.70…

                                                         M 51+
Thai Basil  [12970.399999243304, 182.86941524192605, 48.40…
Ttoust      [10336.143076782753, 69.99744629408036, 41.843…
IB          [12045.958131482006, 75.22744234089875, 83.184…
Poke Bar    [12970.399996687098, 102.05234753231174, 73.92…
```

```
[22]: # table of minimum cost diet constraining nutrients by age-sex groups and␣
      ↪restaurants
      # saved for reference
```

```
excess_tbl
```

[22]:

```
                                                         C 1-3  \
Thai Basil  [Calcium, Ca, Carbohydrate, by difference, Vit…
Ttoust      [Calcium, Ca, Potassium, K, Vitamin B-12, Vita…
IB          [Calcium, Ca, Carbohydrate, by difference, Iro…
Poke Bar    [Carbohydrate, by difference, Potassium, K, Vi…


                                                         F 4-8  \
Thai Basil  [Calcium, Ca, Carbohydrate, by difference, Vit…
Ttoust      [Calcium, Ca, Potassium, K, Vitamin B-12, Vita…
IB          [Calcium, Ca, Carbohydrate, by difference, Iro…
Poke Bar    [Carbohydrate, by difference, Potassium, K, Vi…


                                                         M 4-8  \
Thai Basil  [Calcium, Ca, Carbohydrate, by difference, Vit…
Ttoust      [Calcium, Ca, Potassium, K, Vitamin B-12, Vita…
IB          [Calcium, Ca, Carbohydrate, by difference, Iro…
Poke Bar    [Carbohydrate, by difference, Potassium, K, Vi…


                                                        F 9-13  \
Thai Basil  [Calcium, Ca, Carbohydrate, by difference, Vit…
Ttoust      [Calcium, Ca, Potassium, K, Vitamin B-12, Vita…
IB          [Calcium, Ca, Carbohydrate, by difference, Iro…
Poke Bar    [Carbohydrate, by difference, Phosphorus, P, P…


                                                        M 9-13  \
Thai Basil  [Calcium, Ca, Carbohydrate, by difference, Pot…
Ttoust      [Calcium, Ca, Potassium, K, Vitamin B-12, Vita…
IB          [Calcium, Ca, Carbohydrate, by difference, Iro…
Poke Bar    [Carbohydrate, by difference, Vitamin E (alpha…


                                                       F 14-18  \
Thai Basil  [Calcium, Ca, Carbohydrate, by difference, Pot…
Ttoust      [Calcium, Ca, Potassium, K, Vitamin B-12, Vita…
IB                        [Calcium, Ca, Iron, Fe, Sodium, Na]
Poke Bar    [Carbohydrate, by difference, Vitamin E (alpha…


                                                       M 14-18  \
Thai Basil  [Calcium, Ca, Carbohydrate, by difference, Pot…
Ttoust      [Calcium, Ca, Niacin, Potassium, K, Vitamin B-…
IB          [Calcium, Ca, Carbohydrate, by difference, Iro…
Poke Bar    [Carbohydrate, by difference, Vitamin B-12, Vi…


                                                       F 19-30  \
Thai Basil  [Calcium, Ca, Carbohydrate, by difference, Pot…
Ttoust      [Calcium, Ca, Potassium, K, Vitamin B-12, Vita…
```

```
IB                [Calcium, Ca, Iron, Fe, Sodium, Na, Energy]
Poke Bar    [Carbohydrate, by difference, Potassium, K, Vi…

                                                      M 19-30  \
Thai Basil  [Calcium, Ca, Carbohydrate, by difference, Vit…
Ttoust      [Calcium, Ca, Potassium, K, Vitamin B-12, Vita…
IB          [Carbohydrate, by difference, Magnesium, Mg, T…
Poke Bar    [Carbohydrate, by difference, Vitamin B-12, Vi…

                                                      F 31-50  \
Thai Basil  [Calcium, Ca, Carbohydrate, by difference, Vit…
Ttoust      [Calcium, Ca, Potassium, K, Vitamin B-12, Vita…
IB                [Calcium, Ca, Iron, Fe, Sodium, Na, Energy]
Poke Bar    [Carbohydrate, by difference, Vitamin E (alpha…

                                                      M 31-50  \
Thai Basil  [Calcium, Ca, Carbohydrate, by difference, Pot…
Ttoust      [Calcium, Ca, Potassium, K, Vitamin B-12, Vita…
IB          [Carbohydrate, by difference, Magnesium, Mg, T…
Poke Bar    [Carbohydrate, by difference, Vitamin B-12, Vi…

                                                        F 51+  \
Thai Basil  [Calcium, Ca, Carbohydrate, by difference, Pot…
Ttoust      [Energy, Calcium, Ca, Potassium, K, Vitamin B-…
IB          [Calcium, Ca, Carbohydrate, by difference, Thi…
Poke Bar    [Carbohydrate, by difference, Potassium, K, Vi…

                                                        M 51+
Thai Basil  [Calcium, Ca, Carbohydrate, by difference, Pot…
Ttoust      [Calcium, Ca, Potassium, K, Vitamin B-12, Vita…
IB          [Carbohydrate, by difference, Magnesium, Mg, T…
Poke Bar    [Carbohydrate, by difference, Vitamin B-12, Vi…
```

## 1.5 Sensitivity of solution

### 1.5.1 Effects of Price Changes on Subsistence Diet Cost

[23]:

```python
# examine effects of price change on subsistence diet cost for T-toust, the cheapest option
# for men and women aged 19-30
import cufflinks as cf
cf.go_offline()

group_pc = ['F 19-30', 'M 19-30']

for group in group_pc:
    scale = [.5,.6,.7,.8,.9,1.,1.1,1.2,1.3,1.4,1.5]
    cost0 =␣
↪solve_subsistence_problem(ing_res[1],ing_prices[1],dri_min[group],dri_max[group],tol=tol).
↪fun

    Price_response={}
    for s in scale:
        cost = {}
        for i,p in enumerate(ing_prices[1]):
            my_p = ing_prices[1].copy()
            my_p[i] = p*s
            result =␣
↪solve_subsistence_problem(ing_res[1],my_p,dri_min[group],dri_max[group],tol=tol)
            cost[ing_prices[1].index[i]] = np.log(result.fun/cost0)
        Price_response[np.log(s)] = cost

    Price_response = pd.DataFrame(Price_response).T
    Price_response.iplot(xTitle='change in log price',yTitle='change in log␣
↪cost',
                         title ='Effects of Price Changes on Subsistence Diet␣
↪Cost for '+ group + ' (T-Toust)')
```

```
/opt/conda/lib/python3.9/site-packages/geopandas/_compat.py:111: UserWarning:

The Shapely GEOS version (3.10.3-CAPI-1.16.1) is incompatible with the GEOS
version PyGEOS was compiled with (3.10.4-CAPI-1.16.2). Conversions between both
will be slow.


/tmp/ipykernel_80/2046845037.py:22: UserWarning:

Prices have no units.  BE CAREFUL!  We're assuming prices are per hectogram or
deciliter!
```

### 1.5.2 Effects of Price Changes on Subsistence Diet Composition

[24]:

```
# examine effects of price change on subsistence diet cost for T-toust, the␣
 ↪cheapest option
# for men and women aged 19-30
import cufflinks as cf
cf.go_offline()

ReferenceGood = 'Egg'

group_pc = ['F 19-30', 'M 19-30']
```

```
for group in group_pc:

    scale = [0.5,0.75,0.9,1.,1.1,1.2,1.3,1.4,1.5,2,4]

    cost0 =␣
 ↪solve_subsistence_problem(ing_res[1],ing_prices[1],dri_min[group],dri_max[group],tol=tol).
 ↪fun

    my_p = ing_prices[1].copy()
    diet = {}
    for s in scale:
        my_p[ReferenceGood] = ing_prices[1][ReferenceGood]*s
        result =␣
 ↪solve_subsistence_problem(ing_res[1],my_p,dri_min[group],dri_max[group],tol=tol)
        diet[my_p[ReferenceGood]] = result.diet

    Diet_response = pd.DataFrame(diet).T
    Diet_response.index.name = '%s Price' % ReferenceGood

    Diet_response.reset_index(inplace=True)

    # Get rid of units for index (cufflinks chokes)
    # Diet_response['%s Price' % ReferenceGood] = Diet_response['%s Price' %␣
 ↪ReferenceGood].apply(lambda x: x.magnitude)

    Diet_response = Diet_response.set_index('%s Price' % ReferenceGood)

    # Just look at goods consumed in quantities greater than error tolerance
    Diet_response.loc[:,(Diet_response>tol).sum()>0].iplot(xTitle='%s Price' %␣
 ↪ReferenceGood,yTitle='Hectograms',
                                                          title='Effects of␣
 ↪Price Changes of Eggs on Subsistence Diet Composition for '+group + '␣
 ↪(T-Toust)')
```

```
/tmp/ipykernel_80/2046845037.py:22: UserWarning:

Prices have no units.  BE CAREFUL!  We're assuming prices are per hectogram or
deciliter!
```

## 1.6 Total Cost for Population of Interest

Our population of interest is all UC berkeley students and we assume they are males and females from 19-30. Based on data from UC Berkeley Office of Planning and Analysis(https://opa.berkeley.edu/campus-data/uc-berkeley-quick-facts), there are 23,974 self-reported female students and 20,642 self-reported male students enrolled in Berkeley for Fall 2022.

[25]:
```
#Cost per capita for female and male student
cost_f_1930 = cost_tbl.loc[:,'F 19-30'].min()
cost_m_1930 = cost_tbl.loc[:,'M 19-30'].min()

num_f = 23974
num_m = 20642

#Calculation of total cost for all students
total_cost = cost_f_1930*num_f + cost_m_1930*num_m
total_cost
```

[25]:
```
240099.02962844836
```

### 1.6.1 Total Food Required for Population of Interest

Still, we consider to feed the population by Ttoust, our cheapest option.

[26]:
```
# retrieve saved diet compositions for male and female 19-30
diet_m = diet_tbl.loc['Ttoust', 'M 19-30']
diet_m_name = diet_tbl_name.loc['Ttoust', 'M 19-30']
diet_f = diet_tbl.loc['Ttoust', 'F 19-30']
diet_f_name = diet_tbl_name.loc['Ttoust', 'F 19-30']
```

[28]:
```
# list: Egg, Purple Cabbage, Spinach, Pork, Flour
# Total
egg = diet_m[0]*num_m + diet_f[0]*num_f
pc = diet_m[1]*num_m + diet_f[1]*num_f
spinach = diet_m[2]*num_m + diet_f[2]*num_f
pork = diet_m[3]*num_m + diet_f[3]*num_f
flour = diet_m[4]*num_m + diet_f[4]*num_f
```

[29]:
```
print('We need', egg/10, 'kg of eggs to feed all UC Berkeley students for one␣
  ↪day')
print('We need', pc/10, 'kg of purple cabbage to feed all UC Berkeley students␣
  ↪for one day')
print('We need', spinach/10, 'kg of spinach to feed all UC Berkeley students␣
  ↪for one day')
print('We need', pork/10, 'kg of pork to feed all UC Berkeley students for one␣
  ↪day')
print('We need', flour/10, 'kg of flour to feed all UC Berkeley students for␣
  ↪one day')
```

```
We need 3699.80369851132 kg of eggs to feed all UC Berkeley students for one day
We need 1678.0001626036296 kg of purple cabbage to feed all UC Berkeley students
for one day
We need 27544.590427572606 kg of spinach to feed all UC Berkeley students for
one day
```

```
We need 10349.313552999585 kg of pork to feed all UC Berkeley students for one
day
We need 9622.566477398674 kg of flour to feed all UC Berkeley students for one
day
```