Engel_Project_3_Data_Cleaning_Mapping_Nutrients

April 4, 2022

[194]:
```
import pandas as pd
!pip install pyarrow
```

```
Requirement already satisfied: pyarrow in /opt/conda/lib/python3.9/site-packages
(7.0.0)
Requirement already satisfied: numpy>=1.16.6 in /opt/conda/lib/python3.9/site-
packages (from pyarrow) (1.21.5)
```

[195]:
```
!pip install eep153_tools --upgrade
```

```
Requirement already up-to-date: eep153_tools in /opt/conda/lib/python3.9/site-
packages (0.11)
```

[196]:
```
!pip install -r requirements.txt
```

```
Requirement already satisfied: numpy>=1.20.3 in /opt/conda/lib/python3.9/site-
packages (from -r requirements.txt (line 4)) (1.21.5)
Requirement already satisfied: pandas>=1.2.5 in /opt/conda/lib/python3.9/site-
packages (from -r requirements.txt (line 7)) (1.3.5)
Requirement already satisfied: pint>=0.18 in /opt/conda/lib/python3.9/site-
packages (from -r requirements.txt (line 10)) (0.19)
Requirement already satisfied: requests>=2.26.0 in
/opt/conda/lib/python3.9/site-packages (from -r requirements.txt (line 13))
(2.26.0)
Requirement already satisfied: eep153_tools in /opt/conda/lib/python3.9/site-
packages (from -r requirements.txt (line 15)) (0.11)
Requirement already satisfied: gnupg in /opt/conda/lib/python3.9/site-packages
(from -r requirements.txt (line 17)) (2.3.1)
Requirement already satisfied: python-dateutil>=2.7.3 in
/opt/conda/lib/python3.9/site-packages (from pandas>=1.2.5->-r requirements.txt
(line 7)) (2.8.0)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/lib/python3.9/site-
packages (from pandas>=1.2.5->-r requirements.txt (line 7)) (2021.1)
Requirement already satisfied: idna<4,>=2.5; python_version >= "3" in
/opt/conda/lib/python3.9/site-packages (from requests>=2.26.0->-r
requirements.txt (line 13)) (2.8)
Requirement already satisfied: charset-normalizer~=2.0.0; python_version >= "3"
in /opt/conda/lib/python3.9/site-packages (from requests>=2.26.0->-r
requirements.txt (line 13)) (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 13)) (1.25.7)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/lib/python3.9/site-packages (from requests>=2.26.0->-r
requirements.txt (line 13)) (2019.11.28)
Requirement already satisfied: psutil>=1.2.1 in /opt/conda/lib/python3.9/site-
packages (from gnupg->-r requirements.txt (line 17)) (5.9.0)
Requirement already satisfied: six>=1.5 in /opt/conda/lib/python3.9/site-
packages (from python-dateutil>=2.7.3->pandas>=1.2.5->-r requirements.txt (line
7)) (1.16.0)
```

[197]:

```python
from  scipy.optimize import linprog as lp
import numpy as np
import warnings
from eep153_tools.sheets import read_sheets
```

[198]:

```python
import numpy as np
```

[199]:

```python
z = pd.read_parquet('z.parquet', engine='pyarrow')
```

[200]:

```python
z
```

[200]:

```
k          rural                m      religion          social group  \
j
410001101  Urban          Gujarat      Hinduism  Other backward class
410001102  Urban          Gujarat  Christianity                Others
410001103  Urban          Gujarat      Hinduism                Others
410001201  Urban          Gujarat  Christianity                Others
410001202  Urban          Gujarat      Hinduism                Others
…          …              …           …                   …
799981301  Rural  Jammu & Kashmir      Hinduism                Others
799982101  Rural  Jammu & Kashmir      Hinduism                Others
799982201  Rural  Jammu & Kashmir      Hinduism                Others
799982202  Rural  Jammu & Kashmir      Hinduism                Others
799982301  Rural  Jammu & Kashmir      Hinduism                Others

k          Males 0-1  Males 1-5  Males 5-10  Males 10-15  Males 15-20  \
j
410001101          0          0           0            0            0
410001102          0          0           0            1            0
410001103          0          0           0            0            0
410001201          0          0           0            0            0
410001202          0          0           0            0            0
…              …        …         …          …          …
799981301          0          0           0            1            1
```

```
799982101          0          0           0            1            1
799982201          0          0           0            1            2
799982202          0          0           2            1            0
799982301          0          0           0            0            0

k          Males 20-30  …  Males 60-100  Females 0-1  Females 1-5  \
j                       …
410001101            2  …             0            0            0
410001102            0  …             0            0            0
410001103            3  …             0            0            0
410001201            1  …             1            0            0
410001202            0  …             0            0            1
…                …  …           …          …          …
799981301            0  …             0            0            0
799982101            0  …             0            0            0
799982201            0  …             0            0            0
799982202            0  …             0            0            0
799982301            0  …             1            0            0

k          Females 5-10  Females 10-15  Females 15-20  Females 20-30  \
j
410001101             0              0              0              1
410001102             0              0              0              0
410001103             0              0              0              0
410001201             0              0              0              1
410001202             1              0              0              1
…                 …            …            …            …
799981301             1              1              0              0
799982101             1              0              0              0
799982201             0              1              0              0
799982202             0              0              0              0
799982301             0              0              1              2

k          Females 30-50  Females 50-60  Females 60-100
j
410001101              1              0               0
410001102              1              0               0
410001103              1              0               0
410001201              0              0               0
410001202              0              0               0
…                  …            …             …
799981301              1              0               0
799982101              1              0               0
799982201              1              0               1
799982202              1              0               0
799982301              1              1               0
```

```
[101662 rows x 22 columns]
```

[201]:

```
z.info()
z.m.value_counts()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 101662 entries, 410001101 to 799982301
Data columns (total 22 columns):
 #   Column          Non-Null Count   Dtype 
---  ------          --------------   ----- 
 0   rural           101662 non-null  object
 1   m               101662 non-null  object
 2   religion        101659 non-null  object
 3   social group    101648 non-null  object
 4   Males 0-1       101662 non-null  int64 
 5   Males 1-5       101662 non-null  int64 
 6   Males 5-10      101662 non-null  int64 
 7   Males 10-15     101662 non-null  int64 
 8   Males 15-20     101662 non-null  int64 
 9   Males 20-30     101662 non-null  int64 
 10  Males 30-50     101662 non-null  int64 
 11  Males 50-60     101662 non-null  int64 
 12  Males 60-100    101662 non-null  int64 
 13  Females 0-1     101662 non-null  int64 
 14  Females 1-5     101662 non-null  int64 
 15  Females 5-10    101662 non-null  int64 
 16  Females 10-15   101662 non-null  int64 
 17  Females 15-20   101662 non-null  int64 
 18  Females 20-30   101662 non-null  int64 
 19  Females 30-50   101662 non-null  int64 
 20  Females 50-60   101662 non-null  int64 
 21  Females 60-100  101662 non-null  int64 
dtypes: int64(18), object(4)
memory usage: 17.8+ MB
```

[201]:

```
Uttar Pradesh          9015
Maharashtra            8043
Andhra Pradesh         6899
Tamil Nadu             6647
West Bengal            6315
Madhya Pradesh         4717
Bihar                  4582
Kerala                 4459
Rajasthan              4128
Karnataka              4094
Orissa                 4026
Assam                  3440
```

```
Gujarat                 3426
Jammu & Kashmir         3383
Punjab                  3118
Jharkhand               2740
Haryana                 2591
Manipur                 2560
Chhattisgarh            2169
Himachal Pradesh        2041
Tripura                 1856
Uttaranchal             1783
Arunachal Pradesh       1680
Mizoram                 1536
Meghalaya               1259
Nagaland                1024
Delhi                    951
Sikkim                   768
Pondicherry              576
A & N Islands            566
Goa                      447
Chandigarh               312
Dadra & Nagar Haveli     192
Lakshadweep              191
Daman & Diu              128
Name: m, dtype: int64
```

[202]:

```
z_maha = z[z['m'] == 'Maharashtra']
z_maha
```

[202]:

```
k          rural            m  religion          social group  Males 0-1  \
j
421001201  Urban  Maharashtra  Hinduism                Others          0
421001202  Urban  Maharashtra  Hinduism                Others          0
421001203  Urban  Maharashtra  Hinduism                Others          0
421001204  Urban  Maharashtra  Hinduism  Other backward class          0
421002201  Urban  Maharashtra  Hinduism                Others          0
…          …          …       …                    …        …
756991202  Rural  Maharashtra  Buddhism        Scheduled caste          0
756991203  Rural  Maharashtra  Buddhism        Scheduled caste          0
756991204  Rural  Maharashtra  Buddhism        Scheduled caste          0
756991301  Rural  Maharashtra  Hinduism  Other backward class          0
756991302  Rural  Maharashtra  Hinduism  Other backward class          0

k          Males 1-5  Males 5-10  Males 10-15  Males 15-20  Males 20-30  …  \
j                                                                        …
421001201          1           1            0            0            0  …
421001202          0           0            0            0            0  …
421001203          0           0            0            0            1  …
```

```
421001204          0           0            0            0            0  …
421002201          1           0            0            0            0  …
…              …         …          …          …          …  …
756991202          0           0            0            0            0  …
756991203          0           0            0            0            0  …
756991204          0           0            0            0            0  …
756991301          0           0            0            0            1  …
756991302          0           0            0            1            0  …

k          Males 60-100  Females 0-1  Females 1-5  Females 5-10  \
j
421001201             0            0            0             0
421001202             0            0            1             0
421001203             0            0            0             0
421001204             0            0            0             2
421002201             0            0            0             1
…                 …          …          …           …
756991202             0            0            0             0
756991203             1            0            0             0
756991204             0            1            1             0
756991301             0            0            0             0
756991302             0            0            0             0

k          Females 10-15  Females 15-20  Females 20-30  Females 30-50  \
j
421001201              0              0              1              0
421001202              0              0              0              1
421001203              0              0              0              0
421001204              0              0              0              1
421002201              0              0              0              1
…                  …            …            …            …
756991202              0              0              1              0
756991203              0              0              0              0
756991204              0              0              1              0
756991301              0              1              0              0
756991302              0              1              1              1

k          Females 50-60  Females 60-100
j
421001201              0               0
421001202              0               0
421001203              0               0
421001204              0               0
421002201              0               0
…                  …             …
756991202              0               0
756991203              0               1
```

```
756991204              0                0
756991301              1                1
756991302              0                0

[8043 rows x 22 columns]
```

```
[203]: maha_households = z_maha.index
       maha_households
```

```
[203]: Index(['421001201', '421001202', '421001203', '421001204', '421002201',
              '421002202', '421002203', '421002204', '421011101', '421011102',
              …
              '756982202', '756982301', '756991101', '756991102', '756991201',
              '756991202', '756991203', '756991204', '756991301', '756991302'],
             dtype='object', name='j', length=8043)
```

```
[204]: x = pd.read_parquet('x.parquet', engine='pyarrow').unstack('i')
```

```
[205]: x.index
```

```
[205]: MultiIndex([('410001101', 'Monthly'),
                   ('410001102', 'Monthly'),
                   ('410001103', 'Monthly'),
                   ('410001201', 'Monthly'),
                   ('410001202', 'Monthly'),
                   ('410001203', 'Monthly'),
                   ('410001204', 'Monthly'),
                   ('410001301', 'Monthly'),
                   ('410011101', 'Monthly'),
                   ('410011102', 'Monthly'),
                   …
                   ('799971301', 'Monthly'),
                   ('799971302', 'Monthly'),
                   ('799981101', 'Monthly'),
                   ('799981201', 'Monthly'),
                   ('799981202', 'Monthly'),
                   ('799981301', 'Monthly'),
                   ('799982101', 'Monthly'),
                   ('799982201', 'Monthly'),
                   ('799982202', 'Monthly'),
                   ('799982301', 'Monthly')],
                  names=['j', 'Frequency'], length=101660)
```

```
[206]: x = x.reset_index()
       x
```

```
[206]:                 j Frequency total_value                        \
       i                                 apple arhar (tur) baby food
       0       410001101   Monthly        20.0       121.0       NaN
       1       410001102   Monthly       160.0        60.0       NaN
       2       410001103   Monthly        40.0       195.0       NaN
       3       410001201   Monthly        40.0       130.0       NaN
       4       410001202   Monthly         NaN        65.0       NaN
       …           …       …         …         …       …
       101655  799981301   Monthly         NaN         NaN       NaN
       101656  799982101   Monthly         NaN         NaN       NaN
       101657  799982201   Monthly         NaN         NaN       NaN
       101658  799982202   Monthly         NaN         NaN       NaN
       101659  799982301   Monthly         NaN         NaN       NaN

                                                                    …              \
       i      bajra & products banana barley & products beef beer   … toddy tomato
       0                   NaN    NaN               NaN  NaN  NaN   …   NaN   50.0
       1                  40.0   60.0               NaN  NaN  NaN   …   NaN   12.0
       2                   NaN   50.0               NaN  NaN  NaN   …   NaN   50.0
       3                   NaN   20.0               NaN  NaN  NaN   …   NaN   36.0
       4                   NaN    NaN               NaN  NaN  NaN   …   NaN   30.0
       …                  …    …               …  …  …  …   …    …
       101655              NaN    NaN               NaN  NaN  NaN   …   NaN   30.0
       101656              NaN    NaN               NaN  NaN  NaN   …   NaN   40.0
       101657              NaN    NaN               NaN  NaN  NaN   …   NaN   30.0
       101658              NaN    NaN               NaN  NaN  NaN   …   NaN   20.0
       101659              NaN    NaN               NaN  NaN  NaN   …   NaN   30.0

                                                                     \
       i      turmeric   urd vanaspati, margarine walnut watermelon
       0           7.0   NaN                  NaN    NaN        NaN
       1          20.0  15.0                  NaN   90.0        NaN
       2          12.0   NaN                  NaN    NaN        NaN
       3          33.0   NaN                  NaN  153.0        NaN
       4          10.0   NaN                  NaN    NaN        NaN
       …          …   …                  …    …        …
       101655     20.0   NaN                  NaN    NaN        NaN
       101656     63.0  60.0                  NaN    NaN        NaN
       101657     63.0  60.0                  NaN    NaN        NaN
       101658     63.0  60.0                  NaN    NaN        NaN
       101659     75.0  60.0                  NaN    NaN        NaN


       i      wheat/atta - P.D.S. wheat/atta - other sources zarda,kimam,surti
       0                      NaN                      720.0               NaN
       1                      NaN                      180.0               NaN
       2                      NaN                      600.0               NaN
```

```
3                      NaN                      350.0               NaN
4                      NaN                      220.0               NaN
…                    …                        …               …
101655                 NaN                      160.0               NaN
101656                75.0                      240.0               NaN
101657                60.0                      120.0               NaN
101658                75.0                      180.0               NaN
101659                75.0                      240.0               NaN

[101660 rows x 166 columns]
```

[207]:

```
x_maha = x[x['j'].isin(maha_households)]
x_maha
```

[207]:

```
               j Frequency total_value                                          \
i                                apple arhar (tur) baby food bajra & products
7577   421001201   Monthly         NaN        75.0       NaN              NaN
7578   421001202   Monthly         NaN        80.0       NaN              NaN
7579   421001203   Monthly         NaN         NaN       NaN              NaN
7580   421001204   Monthly         NaN         NaN       NaN              NaN
7581   421002201   Monthly         NaN        75.0       NaN              NaN
…          …       …         …         …       …              …
78734  756991202   Monthly         NaN        30.0       NaN              NaN
78735  756991203   Monthly         NaN       110.0       NaN              NaN
78736  756991204   Monthly         NaN       125.0       NaN              NaN
78737  756991301   Monthly         NaN       130.0       NaN              NaN
78738  756991302   Monthly         NaN        97.0       NaN              NaN

                                          …                            \
i     banana barley & products beef beer  … toddy tomato turmeric urd
7577     NaN               NaN  NaN  NaN  …   NaN   50.0     12.0 NaN
7578    70.0               NaN  NaN  NaN  …   NaN   50.0     12.0 NaN
7579    18.0               NaN  NaN  NaN  …   NaN    NaN      NaN NaN
7580    35.0               NaN  NaN  NaN  …   NaN   50.0     12.0 NaN
7581    35.0               NaN  NaN  NaN  …   NaN   50.0     12.0 NaN
…      …               …  …  …  …   …    …      …  . .
78734   40.0               NaN  NaN  NaN  …   NaN   35.0     18.0 NaN
78735    NaN               NaN  NaN  NaN  …   NaN   12.0     12.0 NaN
78736    NaN               NaN  NaN  NaN  …   NaN    6.0     18.0 NaN
78737   22.0               NaN  NaN  NaN  …   NaN   25.0     18.0 NaN
78738    NaN               NaN  NaN  NaN  …   NaN   35.0     30.0 NaN

                                                                  \
i     vanaspati, margarine walnut watermelon wheat/atta - P.D.S.
7577                   NaN    NaN        NaN                NaN
7578                   NaN    NaN        NaN                NaN
7579                   NaN    NaN        NaN                NaN
```

```
7580                   NaN    NaN        NaN                 NaN
7581                   NaN    NaN        NaN                 NaN
…                    …    …        …                 …
78734                  NaN    NaN        NaN                 NaN
78735                  NaN    NaN        NaN                40.0
78736                  NaN    NaN        NaN               160.0
78737                  NaN    NaN        NaN                50.0
78738                  NaN    NaN        NaN                 NaN


i     wheat/atta - other sources zarda,kimam,surti
7577                       400.0               NaN
7578                       120.0               NaN
7579                         NaN               NaN
7580                       480.0               NaN
7581                       400.0               NaN
…                          …               …
78734                      140.0               NaN
78735                        NaN               NaN
78736                        NaN               NaN
78737                        NaN               NaN
78738                        NaN               NaN

[8043 rows x 166 columns]
```

[208]:
```
x_maha.columns.values.tolist()
```

[208]:
```
[('j', ''),
 ('Frequency', ''),
 ('total_value', 'apple'),
 ('total_value', 'arhar (tur)'),
 ('total_value', 'baby food'),
 ('total_value', 'bajra & products'),
 ('total_value', 'banana'),
 ('total_value', 'barley & products'),
 ('total_value', 'beef'),
 ('total_value', 'beer'),
 ('total_value', 'berries'),
 ('total_value', 'besan'),
 ('total_value', 'bidi'),
 ('total_value', 'biscuits, chocolates'),
 ('total_value', 'black pepper'),
 ('total_value', 'bread (bakery)'),
 ('total_value', 'brinjal'),
 ('total_value', 'butter'),
 ('total_value', 'cabbage'),
 ('total_value', 'cake, pastry, prepared sweets'),
 ('total_value', 'candle '),
 ('total_value', 'candy (misri)'),
 ('total_value', 'carrot'),
 ('total_value', 'cashewnut'),
 ('total_value', 'cauliflower'),
 ('total_value', 'cereal substitutes  (tapioca, jackfruit seed etc.)'),
 ('total_value', 'charcoal'),
 ('total_value', 'cheroot '),
 ('total_value', 'chicken'),
 ('total_value', 'chillis (green)'),
 ('total_value', 'chips'),
 ('total_value', 'chira'),
 ('total_value', 'cigarettes '),
 ('total_value', 'coal'),
 ('total_value', 'coconut'),
 ('total_value', 'coconut (copra)'),
 ('total_value', 'coconut oil'),
 ('total_value', 'coconut: green'),
 ('total_value', 'coffee : cups'),
 ('total_value', 'coffee: powder'),
 ('total_value', 'coke '),
 ('total_value', 'cold beverages: bottled/canned'),
 ('total_value', 'cooked meals'),
 ('total_value', 'cooked snacks purchased [samosa, puri, paratha,'),
 ('total_value', 'country liquor'),
 ('total_value', 'curd'),
 ('total_value', 'curry powder'),
 ('total_value', 'dates'),
 ('total_value', 'dhania'),
 ('total_value', 'diesel '),
 ('total_value', 'dry chillies'),
 ('total_value', 'dung cake'),
 ('total_value', 'edible oil (others)'),
 ('total_value', 'eggs'),
 ('total_value', 'electricity '),
 ('total_value', 'firewood & chips'),
 ('total_value', 'fish ( fresh )'),
 ('total_value', 'foreign liquor or refined liquor'),
 ('total_value', 'french beans and barbati'),
 ('total_value', 'fruit juice and shake'),
 ('total_value', 'ganja '),
 ('total_value', 'garlic'),
 ('total_value', 'ghee'),
 ('total_value', 'ginger'),
 ('total_value', 'goat meat'),
 ('total_value', 'gobar gas'),
 ('total_value', 'gourd, pumpkin'),
 ('total_value', 'gram (split)'),
 ('total_value', 'gram (whole)'),
 ('total_value', 'gram products'),
 ('total_value', 'grapes'),
 ('total_value', 'groundnut'),
 ('total_value', 'groundnut oil'),
 ('total_value', 'guava'),
 ('total_value', 'gur'),
 ('total_value', 'honey'),
 ('total_value', 'hooka tobacco'),
 ('total_value', 'ice-cream'),
 ('total_value', 'ingredients for pan'),
 ('total_value', 'jackfruit'),
 ('total_value', 'jeera'),
 ('total_value', 'jowar & products'),
 ('total_value', 'kerosene-other sources'),
 ('total_value', 'kerosene-pds '),
 ('total_value', 'kharbooza'),
 ('total_value', 'khesari'),
 ('total_value', 'khoi, lawa'),
 ('total_value', "lady's finger"),
 ('total_value', 'leaf tobacco'),
 ('total_value', 'leechi'),
 ('total_value', 'lemon'),
 ('total_value', 'lpg'),
 ('total_value', 'maida'),
 ('total_value', 'maize & products'),
 ('total_value', 'mango'),
 ('total_value', 'masur'),
 ('total_value', 'matches '),
 ('total_value', 'milk : condensed/ powder'),
 ('total_value', 'milk: liquid'),
 ('total_value', 'mineral water'),
 ('total_value', 'moong'),
 ('total_value', 'muri'),
 ('total_value', 'mustard oil'),
 ('total_value', 'oilseeds'),
 ('total_value', 'onion'),
 ('total_value', 'orange,mausami'),
 ('total_value', 'other beverages (cocoa, chocolate etc.)'),
 ('total_value', 'other cereals'),
 ('total_value', 'other dry fruits'),
 ('total_value', 'other fresh fruits'),
 ('total_value', 'other fuel '),
 ('total_value', 'other intoxicants '),
 ('total_value', 'other milk products'),
 ('total_value', 'other nuts'),
 ('total_value', 'other packaged processed food'),
 ('total_value', 'other pulse products'),
 ('total_value', 'other pulses'),
 ('total_value', 'other rice products'),
 ('total_value', 'other served processed food'),
 ('total_value', 'other spices'),
 ('total_value', 'other tobacco products '),
 ('total_value', 'other vegetables'),
 ('total_value', 'other wheat products'),
 ('total_value', 'others (birds, crab, oyster, tortoise etc.)'),
 ('total_value', 'palak'),
 ('total_value', 'pan : finished'),
 ('total_value', 'pan : leaf'),
 ('total_value', 'papad, bhujia, namkeen, mixture, chanachur'),
 ('total_value', 'papaya'),
 ('total_value', 'parwal / patal'),
 ('total_value', 'pears (naspati)'),
 ('total_value', 'peas-pulses'),
 ('total_value', 'peas-vegetables'),
 ('total_value', 'petrol '),
 ('total_value', 'pickles'),
 ('total_value', 'pineapple'),
 ('total_value', 'pork'),
 ('total_value', 'potato'),
 ('total_value', 'radish'),
 ('total_value', 'ragi & products'),
 ('total_value', 'raisin (kishmish, monacca etc.)'),
 ('total_value', 'refined oil [sunflower, soyabean, saffola, etc.]'),
 ('total_value', 'rice - P.D.S.'),
 ('total_value', 'rice - other sources'),
 ('total_value', 'salt '),
 ('total_value', 'sauce, jam, jelly'),
 ('total_value', 'sewai, noodles'),
 ('total_value', 'singara'),
 ('total_value', 'small millets & products'),
 ('total_value', 'snuff '),
 ('total_value', 'sugar - P.D.S.'),
 ('total_value', 'sugar - other sources'),
 ('total_value', 'suji, rawa'),
 ('total_value', 'tamarind'),
 ('total_value', 'tea : cups'),
 ('total_value', 'tea : leaf'),
 ('total_value', 'toddy'),
 ('total_value', 'tomato'),
 ('total_value', 'turmeric'),
 ('total_value', 'urd'),
 ('total_value', 'vanaspati, margarine'),
 ('total_value', 'walnut'),
 ('total_value', 'watermelon'),
 ('total_value', 'wheat/atta - P.D.S.'),
 ('total_value', 'wheat/atta - other sources'),
 ('total_value', 'zarda,kimam,surti')]
```

[209]:

```
total_expenditures = pd.read_parquet('total_expenditures.parquet',␣
 ↪engine='pyarrow')
```

[210]:

```
total_expenditures
```

[210]:

```
           total_value
j
410001101         7813
410001102         3573
410001103         9359
410001201         5671
410001202         6169
…                  …
799981301         3842
799982101         2736
799982201         3378
799982202         3221
799982301         3777

[101660 rows x 1 columns]
```

[211]:

```
total_expenditures_maha = total_expenditures[total_expenditures.index.
 ↪isin(maha_households)]
total_expenditures_maha
```

[211]:

```
           total_value
j
421001201         4857
421001202         5246
421001203         2725
421001204         4750
421002201         5207
…                  …
756991202         2497
756991203         2028
756991204         2833
756991301         3706
756991302         4566

[8043 rows x 1 columns]
```

[213]:
```
n = pd.read_parquet('n.parquet', engine='pyarrow')
n
```

[213]:
```
     calories per unit(kcal)  fat per unit(gm)  \
1                3280.000000             13.00
4                1100.000000              2.00
5                3420.000000             36.00
7                3420.000000             36.00
8                3360.000000             13.00
..                       …                 …
145                24.700001              0.95
146                21.100000              0.85
147                28.500000              0.17
148                24.700001              0.95
149                24.700001              0.95

                                          i  protein per unit(gm)  rural  \
1                                      ragi                 73.00    NaN
4                         other cereal subs.                16.00    NaN
5                     maize-other sources                  111.00    NaN
7                              maize - pds                 111.00    NaN
8                                   barley                 115.00    NaN
..                                       …                     …     …
145              other served processed food                 0.70    0.0
146            cake, pastry, prepared sweets                 0.20    0.0
147                   biscuits, chocolates                   0.35    0.0
148  papad, bhujia, namkeen, mixture, chanachur              0.70    0.0
149            other packaged processed food                 0.70    0.0

      t unit
1    50   kg
4    50   kg
5    50   kg
7    50   kg
8    50   kg
..   ..  …
145  68   Re
146  68   Re
147  68   Re
148  68   Re
149  68   Re

[277 rows x 7 columns]
```

[214]:
```
#set(n.i.tolist()).intersection(qi)
```

```
[219]: q = pd.read_parquet('q.parquet', engine='pyarrow').reset_index()
       q
```

```
[219]:                  j                        i unit Frequency  total_quantity
       0        410001101                    apple   kg   Monthly           250.0
       1        410001101              arhar (tur)   kg   Monthly          2000.0
       2        410001101                    besan   kg   Monthly          2000.0
       3        410001101             black pepper   gm   Monthly            20.0
       4        410001101                  brinjal   kg   Monthly          5000.0
       …              …                       …   …    …             …
       4423639  799982301                   tomato   kg   Monthly          3000.0
       4423640  799982301                 turmeric   gm   Monthly           300.0
       4423641  799982301                      urd   kg   Monthly          1000.0
       4423642  799982301      wheat/atta - P.D.S.   kg   Monthly         10000.0
       4423643  799982301  wheat/atta - other sources   kg   Monthly         20000.0

       [4423644 rows x 5 columns]
```

```
[188]: #q.get_level('i')
```

```
[216]: qi = q.index.get_level_values('i')
       set(n.i.tolist()).intersection(qi)
```

```
[216]: {'apple',
        'arhar (tur)',
        'baby food',
        'bajra & products',
        'banana',
        'barley & products',
        'beef',
        'beer',
        'berries',
        'besan',
        'biscuits, chocolates',
        'black pepper',
        'bread (bakery)',
        'brinjal',
        'butter',
        'cabbage',
        'cake, pastry, prepared sweets',
        'candy (misri)',
        'carrot',
        'cashewnut',
        'cauliflower',
        'cereal substitutes  (tapioca, jackfruit seed etc.)',
        'chicken',
        'chillis (green)',
        'chips',
        'chira',
        'coconut',
        'coconut (copra)',
        'coconut oil',
        'coconut: green',
        'coffee : cups',
        'coffee: powder',
        'cold beverages: bottled/canned',
        'cooked meals',
        'cooked snacks purchased [samosa, puri, paratha,',
        'country liquor',
        'curd',
        'curry powder',
        'dates',
        'dhania',
        'dry chillies',
        'edible oil (others)',
        'eggs',
        'fish ( fresh )',
        'foreign liquor or refined liquor',
        'french beans and barbati',
        'fruit juice and shake',
        'garlic',
        'ghee',
        'ginger',
        'goat meat',
        'gourd, pumpkin',
        'gram (split)',
        'gram (whole)',
        'gram products',
        'grapes',
        'groundnut',
        'groundnut oil',
        'guava',
        'gur',
        'honey',
        'ice-cream',
        'ingredients for pan',
        'jackfruit',
        'jeera',
        'jowar & products',
        'kharbooza',
        'khesari',
        'khoi, lawa',
        "lady's finger",
        'leechi',
        'lemon',
        'maida',
        'maize & products',
        'mango',
        'masur',
        'milk : condensed/ powder',
        'milk: liquid',
        'moong',
        'muri',
        'mustard oil',
        'oilseeds',
        'onion',
        'orange,mausami',
        'other beverages (cocoa, chocolate etc.)',
        'other cereals',
        'other dry fruits',
        'other fresh fruits',
        'other milk products',
        'other nuts',
        'other packaged processed food',
        'other pulse products',
        'other pulses',
        'other rice products',
        'other served processed food',
        'other spices',
        'other vegetables',
        'other wheat products',
        'others (birds, crab, oyster, tortoise etc.)',
        'palak',
        'pan : finished',
        'pan : leaf',
        'papad, bhujia, namkeen, mixture, chanachur',
        'papaya',
        'parwal / patal',
        'pears (naspati)',
        'peas-vegetables',
        'pickles',
        'pineapple',
        'pork',
        'potato',
        'radish',
        'ragi & products',
        'raisin (kishmish, monacca etc.)',
        'refined oil [sunflower, soyabean, saffola, etc.]',
        'rice - P.D.S.',
        'rice - other sources',
        'sauce, jam, jelly',
        'sewai, noodles',
        'singara',
        'small millets & products',
        'sugar - P.D.S.',
        'sugar - other sources',
        'suji, rawa',
        'tamarind',
        'tea : cups',
        'tea : leaf',
        'toddy',
        'tomato',
        'turmeric',
        'urd',
        'vanaspati, margarine',
        'walnut',
        'watermelon',
        'wheat/atta - P.D.S.',
        'wheat/atta - other sources'}
```

[243]:
```
q_maha = q[q['j'].isin(maha_households)]
#q_maha = q_maha.drop_duplicates(subset=['i'])
q_maha
```

[243]:
```
                 j                    i unit Frequency  total_quantity
332920   421001201          arhar (tur)   kg   Monthly          1000.0
332921   421001201                besan   kg   Monthly           500.0
332922   421001201  biscuits, chocolates   Re   Monthly             0.0
332923   421001201        bread (bakery)   kg   Monthly          1000.0
332924   421001201              brinjal   kg   Monthly          1000.0
…              …                    …   …    …              …
3494160  756991302           suji, rawa   kg   Monthly          1000.0
3494161  756991302           tea : cups  no.   Monthly            20.0
3494162  756991302           tea : leaf   gm   Monthly           350.0
3494163  756991302               tomato   kg   Monthly          3500.0
3494164  756991302             turmeric   gm   Monthly           150.0

[387953 rows x 5 columns]
```

[221]:
```
#fdc_codes = pd.read_csv('proj_3_fdc_codes.csv').set_index('Item')
#fdc_codes.index.name = 'i'
```

[222]:
```
#food_items = fdc_codes['Item'].to_list()
#len(food_items)
```

[226]:
```
#q_maha['i'].unique()
```

[223]:

```
#updated_food_items = []
#for item in q_maha['i'].unique():
  #if item in food_items:
    #updated_food_items.append(item)
#updated_food_items
#len(updated_food_items)
```

[227]:

```
#q_maha = q_maha[q_maha['i'].isin(updated_food_items)]
#q_maha
```

[228]:

```
#fdc_codes = fdc_codes[fdc_codes.index.isin(updated_food_items)]
#fdc_codes = fdc_codes.rename(columns={'Item':'i'})
#fdc_codes
```

[229]:

```
#food_codes()
```

[230]:

```
#new_q_maha = pd.concat([q_maha, fdc_codes], axis=1)
#new_q_maha
```

[231]:

```
#food
#new_q_maha[new_q_maha.i==food,:].ID[0]
```

[232]:

```
#q_maha = q_maha.set_index(['j','i'])['total_quantity'].unstack('i')
#q_maha
```

[238]:

```
N = n.loc[n.t=='68',:].set_index('i').drop(columns=['rural', 't', 'unit'])
N = N.reset_index()
N
```

[238]:

```
                                              i  calories per unit(kcal)  \
0                                 rice - P.D.S.              3460.000000
1                          rice - other sources              3460.000000
2                                         chira              3460.000000
3                                    khoi, lawa              3250.000000
4                                          muri              3250.000000
..                                          …                       …
143                 other served processed food                24.700001
144               cake, pastry, prepared sweets                21.100000
145                        biscuits, chocolates                28.500000
146  papad, bhujia, namkeen, mixture, chanachur                24.700001
147               other packaged processed food                24.700001

     fat per unit(gm)  protein per unit(gm)
0                5.00                 75.00
1                5.00                 75.00
2               12.00                 66.00
3                1.00                 75.00
```

```
4             1.00                 75.00
..             …                   …
143           0.95                  0.70
144           0.85                  0.20
145           0.17                  0.35
146           0.95                  0.70
147           0.95                  0.70

[148 rows x 4 columns]
```

[239]:
```
N = N.drop_duplicates(subset=['i'])
N
```

[239]:
```
                                    i  calories per unit(kcal)  \
0                       rice - P.D.S.                  3460.00
1                rice - other sources                  3460.00
2                               chira                  3460.00
3                          khoi, lawa                  3250.00
4                                muri                  3250.00
..                                 …                    …
133               ingredients for pan                     6.55
134                             toddy                   380.00
135                    country liquor                   380.00
136                              beer                   380.00
137  foreign liquor or refined liquor                   380.00

     fat per unit(gm)  protein per unit(gm)
0                5.00                 75.00
1                5.00                 75.00
2               12.00                 66.00
3                1.00                 75.00
4                1.00                 75.00
..                …                   …
133              0.59                  0.21
134              3.00                  1.00
135              3.00                  1.00
136              3.00                  1.00
137              3.00                  1.00

[136 rows x 4 columns]
```

[261]:
```
new_df = q_maha.merge(N, left_on='i', right_on='i')
new_df
```

[261]:
```
                j                  i unit Frequency  total_quantity  \
0       421001201        arhar (tur)   kg   Monthly          1000.0
1       421001202        arhar (tur)   kg   Monthly          1000.0
```

```
2       421002201        arhar (tur)   kg   Monthly          1000.0
3       421002202        arhar (tur)   kg   Monthly          1000.0
4       421002203        arhar (tur)   kg   Monthly          1000.0
…           …                …  …       …             …
331365  756361201  barley & products   kg   Monthly          2000.0
331366  756361202  barley & products   kg   Monthly          3000.0
331367  756361203  barley & products   kg   Monthly          2000.0
331368  756361204  barley & products   kg   Monthly          4000.0
331369  756361301  barley & products   kg   Monthly         10000.0

        calories per unit(kcal)  fat per unit(gm)  protein per unit(gm)
0                        3350.0              17.0                 223.0
1                        3350.0              17.0                 223.0
2                        3350.0              17.0                 223.0
3                        3350.0              17.0                 223.0
4                        3350.0              17.0                 223.0
…                          …               …                   …
331365                   3360.0              13.0                 115.0
331366                   3360.0              13.0                 115.0
331367                   3360.0              13.0                 115.0
331368                   3360.0              13.0                 115.0
331369                   3360.0              13.0                 115.0

[331370 rows x 8 columns]
```

[266]:

```
fdc_codes = pd.read_csv('proj_3_fdc_codes.csv').set_index('Item')
fdc_codes = fdc_codes.reset_index()
```

[266]:

```
                          Item       ID
0                        apple  1102644
1                  arhar (tur)  1977550
2                    baby food  1102843
3             bajra & products  1799770
4                       banana  1102653
..                          …      …
90                         urd  1898206
91        vanaspati, margarine  1103828
92                      walnut  2118446
93                  watermelon  1102698
94  wheat/atta - other sources   522973

[95 rows x 2 columns]
```

[268]:

```
#this is the final dataframe
new_df_codes = new_df.merge(fdc_codes, left_on='i', right_on='Item')
new_df_codes
```

```
[268]:
                 j                  i unit Frequency  total_quantity  \
0        421001201        arhar (tur)   kg   Monthly          1000.0
1        421001202        arhar (tur)   kg   Monthly          1000.0
2        421002201        arhar (tur)   kg   Monthly          1000.0
3        421002202        arhar (tur)   kg   Monthly          1000.0
4        421002203        arhar (tur)   kg   Monthly          1000.0
…           …                …  …       …             …
233493  756361201  barley & products   kg   Monthly          2000.0
233494  756361202  barley & products   kg   Monthly          3000.0
233495  756361203  barley & products   kg   Monthly          2000.0
233496  756361204  barley & products   kg   Monthly          4000.0
233497  756361301  barley & products   kg   Monthly         10000.0

        calories per unit(kcal)  fat per unit(gm)  protein per unit(gm)  \
0                        3350.0              17.0                 223.0
1                        3350.0              17.0                 223.0
2                        3350.0              17.0                 223.0
3                        3350.0              17.0                 223.0
4                        3350.0              17.0                 223.0
…                         …               …                   …
233493                   3360.0              13.0                 115.0
233494                   3360.0              13.0                 115.0
233495                   3360.0              13.0                 115.0
233496                   3360.0              13.0                 115.0
233497                   3360.0              13.0                 115.0

                     Item       ID
0             arhar (tur)  1977550
1             arhar (tur)  1977550
2             arhar (tur)  1977550
3             arhar (tur)  1977550
4             arhar (tur)  1977550
…                   …      …
233493  barley & products  2072684
233494  barley & products  2072684
233495  barley & products  2072684
233496  barley & products  2072684
233497  barley & products  2072684

[233498 rows x 10 columns]
```

```
[ ]:
import fooddatacentral as fdc
apikey = 'CDXgPa1HVqJab8EFllem1ikOF75m2ELYwziKtICr'
D = {}
count = 0
for food in new_df_codes.i.tolist():
    try:
        FDC = new_df_codes.loc[new_df_codes.i==food,:].ID[count]
        count+=1
        D[food] = fdc.nutrients(apikey,FDC).total_quantity
    except AttributeError:
        warnings.warn("Couldn't find FDC Code %s for food %s." % (food,FDC))

D = pd.DataFrame(D,dtype=float).fillna(0)

D
```

```
/tmp/ipykernel_58/2604839490.py:11: UserWarning: Couldn't find FDC Code arhar
(tur) for food 1977550.
  warnings.warn("Couldn't find FDC Code %s for food %s." % (food,FDC))
```

[245]:
```
#idx = N.index.drop_duplicates().intersection(q_maha.columns)
#N = N.loc[idx,:]
```

[246]:
```
#N.iloc[:,:3].loc[idx,:].index.drop_duplicates()
```

[247]:
```
#values = N.index.duplicated()
#duplicated_vals = []
#for i in range(len(values)):
   #if values[i] == True:
        #duplicated_vals.append(i)
#duplicated_vals
```

[248]:
```
#N = N.drop(N.index[duplicated_vals])
#N
```

[250]:
```
#idx = N.index.intersection(q_maha.columns)
#N = N.loc[idx,:]
#N
```

[251]:
```
#q_maha = q_maha.drop(columns=['ice-cream', 'mineral water', 'peas-vegetables',␣
 ↪'potato'])
#q_maha
```

[252]:
```
#N = N.sort_index(ascending=True)
#N.index
```

[253]:
```
#N = N.reset_index()
#N
```

[255]:
```
#q_maha = q_maha.reset_index()
#q_maha
```

[256]:

```
#matrix_mult = q_maha@N
#matrix_mult
```

[257]:

```
#matrix_mult.isnull().sum().sum()
```

[ ]:

```

```