```
  Preparing metadata (setup.py) … done
Requirement already satisfied: psutil>=1.2.1 in
/opt/conda/lib/python3.9/site-packages (from gnupg) (5.9.4)
Building wheels for collected packages: gnupg
  Building wheel for gnupg (setup.py) … done
  Created wheel for gnupg: filename=gnupg-2.3.1-py3-none-any.whl
size=94620
sha256=641ce2addf28ea27f2bc62b0048203ea9187e28db6989edd7dae36be556579f3
  Stored in directory: /home/jovyan/.cache/pip/wheels/20/7e/30/7d702acd6a1e89911
301cd9dbf9cb9870ca80c0e64bc2cde23
Successfully built gnupg
Installing collected packages: gnupg
Successfully installed gnupg-2.3.1
Note: you may need to restart the kernel to use updated packages.
```

```
/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.


Missing dependencies for OracleDemands.
```

2 (A) Choice of Population, with supporting expenditure

We chose to analyze the Ugandan popultion of males and females 19-30

Ugandan Expenditures of 2019-20

[2]:
```
Uganda_Data = '1yVLriVpo7KGUXvR3hq_n53XpXlD5NmLaH1oOMZyV0gQ'

x = read_sheets(Uganda_Data,sheet='Expenditures (2019-20)')
x.columns.name = 'j'
```

```
Key available for students@eep153.iam.gserviceaccount.com.
```

Ugandan Household characteristics

[3]:
```
d = read_sheets(Uganda_Data,sheet="HH Characteristics")
d.columns.name = 'k'
```

```
Key available for students@eep153.iam.gserviceaccount.com.
```

[4]:
```
x = x.groupby('j',axis=1).sum() #reducing duplicate columns
x = x.replace(0,np.nan) #reducing nulls
y = np.log(x.set_index(['i','t','m'])) #log of expenditure
d.set_index(['i','t','m'],inplace=True) #specific labels for the axis
```

```
use = y.index.intersection(d.index)
y = y.loc[use,:]
d = d.loc[use,:]


#Filtering it down to our population of interest (M,F 19-30)
b = read_sheets(Uganda_Data,sheet='RDI')
b = b.set_index('n')
```

```
Key available for students@eep153.iam.gserviceaccount.com.
```

3 (A) Estimate Demand System

[5]:
```
from cfe.estimation import drop_columns_wo_covariance
y = drop_columns_wo_covariance(y,min_obs=30)
use = y.index.intersection(d.index)
y = y.loc[use,:]
d = d.loc[use,:]

#y is log expednitures on food j by household i at a particular time
y = y.stack()
d = d.stack()
assert y.index.names == ['i','t','m','j']
assert d.index.names == ['i','t','m','k']

#setting up the regression
result = Regression(y=y,d=d)
#predicting expenditures
result.predicted_expenditures()
```

[5]:
```
i                                 t        m         j
00c9353d8ebe42faabf5919b81d7fae7  2019-20  Eastern   Beans
3555.677276
                                                     Beef

8401.789558
                                                     Biscuits

842.091521
                                                     Bread

3077.266434
                                                     Cabbages

1199.255865
                                                                          …
e07bc322c4884559b4b8ca75c945dd3e  2019-20  Northern  Sweet Potatoes
6706.688800
                                                     Tea

201.911345
```

```
                                                                Tomatoes
1349.788766
                                                                Waragi
4423.328381
                                                                Yam
2839.871449
Length: 101010, dtype: float64
```

Comparing Predicted Log Expenditures with Actual Expenditures

[6]:
```python
%matplotlib notebook
df = pd.DataFrame({'y':y,'yhat':result.get_predicted_log_expenditures()})
df.plot.scatter(x='yhat',y='y', title = 'Log Expenditures vs. Actual␣
 ↪Expenditures')
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.HTML object>
```

[6]:
```
<AxesSubplot:title={'center':'Log Expenditures vs. Actual Expenditures'},
xlabel='yhat', ylabel='y'>
```

Demand and Household Composition Relative to the average consumption, the characteristics of age and sex affect the demand of the household in this factor.

[7]:
```python
result.gamma
```

[7]:
```
k                  F 00-03   F 04-08   F 09-13   F 14-18   F 19-30   F 31-50  \
j
Beans            -0.124336  0.035231  0.090377  0.004499 -0.007870  0.022907
Beef             -0.133073  0.012580  0.015526  0.014861  0.082507  0.061304
Biscuits          0.038606 -0.000608 -0.026963  0.101941  0.015177  0.244867
Bread            -0.064686  0.027238 -0.092955 -0.008487  0.061894  0.057456
Cabbages          0.007378 -0.058572  0.029930  0.037955 -0.026252  0.036936
Cassava           0.019206  0.089485  0.105922  0.042049 -0.004145  0.072680
Chapati          -0.034054 -0.008517  0.065749  0.090993  0.023017  0.006971
Cooking Oil      -0.088741 -0.050446 -0.052850  0.011923  0.028813  0.017072
Dodo             -0.083900 -0.011246  0.091461  0.040517  0.049264  0.073878
Eggs             -0.086195 -0.056274 -0.008720 -0.024014  0.022208 -0.015094
Fish (dried)     -0.000155 -0.008999  0.021902  0.021847  0.011493  0.045895
Fish (fresh)      0.009838  0.083777  0.125428  0.059686  0.075814  0.085064
Ground Nuts       0.013161  0.014915  0.075040  0.063653  0.080324  0.096478
Kabalagala       -0.032909 -0.051275  0.069242 -0.000100 -0.088485 -0.053790
Maize            -0.064700  0.044877  0.088067  0.076674 -0.019120 -0.016897
Mangos           -0.114765  0.017640  0.028032  0.038333  0.054714 -0.059206
Matoke           -0.032716 -0.035205  0.003864  0.036773 -0.049860  0.046850
Milk (fresh)      0.035783 -0.008844 -0.003371  0.039232  0.068701  0.176070
Millet           -0.121593  0.029121  0.032358 -0.025128  0.066097  0.090670
```

Onions	-0.051784	-0.031392	-0.014769	0.005998	0.087034	-0.026452
Oranges	0.060860	0.110444	0.046275	-0.004837	0.323084	0.149912
Other Fruits	-0.130304	0.039837	0.046928	0.005834	0.019575	0.104073
Other Veg.	-0.050658	-0.001334	0.034464	-0.009238	0.044014	0.111230
Peas	-0.026447	-0.063961	0.052868	-0.013635	-0.088053	0.043005
Pork	0.028176	0.019936	0.055182	-0.167984	0.142273	0.114264
Rice	-0.002402	0.012209	0.018467	0.044520	0.032227	0.060999
Salt	-0.028577	0.015672	0.007313	-0.013734	-0.025414	0.031930
Sim Sim	0.044791	-0.118614	0.030846	0.002150	0.011740	0.015900
Soda	-0.003757	-0.114090	0.028002	0.076334	0.057666	0.057243
Sweet Bananas	0.026802	-0.005293	-0.091636	-0.007828	0.042883	0.134667
Sweet Potatoes	0.078872	0.055466	0.187521	0.037386	-0.002830	0.115568
Tea	-0.009702	-0.032590	0.008430	0.036676	0.072003	0.136950
Tomatoes	-0.028199	-0.039558	-0.037468	0.002304	0.052460	0.029078
Waragi	-0.288787	0.048570	-0.218171	-0.144002	-0.132274	-0.107632
Yam	-0.129158	0.091401	0.026173	0.113589	-0.057574	-0.083954

k	F 51+	M 00-03	M 04-08	M 09-13	M 14-18	M 19-30 \
j						
Beans	0.102685	-0.042510	0.022145	0.064256	0.063150	0.041227
Beef	0.189160	-0.042146	0.021119	-0.009366	0.048574	0.077610
Biscuits	0.289385	0.243284	0.039812	-0.050499	-0.058406	0.046866
Bread	0.084374	-0.128560	0.073478	-0.041311	-0.019199	0.013019
Cabbages	0.081504	-0.045031	0.033217	0.013162	0.042502	0.069609
Cassava	0.156273	-0.003705	0.134056	0.186236	0.135270	0.062836
Chapati	-0.090070	0.007639	0.082807	-0.036377	-0.016584	0.067900
Cooking Oil	-0.079372	-0.086822	-0.083002	-0.041623	-0.007363	-0.034479
Dodo	0.182246	-0.021156	0.021386	0.088145	-0.009156	0.044575
Eggs	0.105724	0.008888	-0.000990	-0.033516	0.009427	-0.022596
Fish (dried)	0.170180	0.001304	0.026337	0.015620	-0.108256	0.056007
Fish (fresh)	0.027070	0.085254	0.044234	0.125689	0.038438	0.096345
Ground Nuts	0.226823	0.046013	0.020437	0.014850	0.049152	0.058602
Kabalagala	-0.011311	-0.101236	-0.183511	-0.026281	-0.063476	0.013625
Maize	0.150861	0.005856	-0.023300	0.085845	0.074174	0.027056
Mangos	-0.008494	0.111844	-0.023910	0.013759	0.040844	-0.046956
Matoke	0.181585	-0.037656	-0.008610	0.003132	0.025605	0.049039
Milk (fresh)	0.243552	0.008782	-0.028640	-0.004941	-0.021904	0.053014
Millet	0.195719	-0.119983	-0.032915	0.106048	0.009796	0.013473
Onions	0.001169	-0.059100	-0.045887	-0.000122	-0.019630	0.011492
Oranges	0.195071	0.056319	0.186683	0.208308	0.151403	0.068689
Other Fruits	0.177591	0.038180	-0.045770	-0.023611	0.044638	0.056425
Other Veg.	0.285344	0.066230	0.015528	0.059022	0.015751	0.061991
Peas	0.071163	-0.030451	-0.134744	0.018332	0.071122	-0.113014
Pork	0.112907	-0.007306	-0.003249	0.035021	-0.042740	0.009620
Rice	0.099859	-0.015938	0.067081	0.058842	0.025188	0.046406
Salt	0.076741	-0.002482	0.004264	0.038216	0.019241	-0.009647
Sim Sim	0.159403	0.072875	0.020773	0.087897	-0.000764	0.096785

Soda	0.032375	0.085763	-0.032433	0.118702	0.025804	0.131296
Sweet Bananas	0.249935	0.157172	0.094374	0.009959	0.054098	0.051740
Sweet Potatoes	0.303346	0.067531	0.094755	0.141835	0.104988	0.039756
Tea	0.207373	-0.061743	-0.010517	0.018140	-0.011002	0.019937
Tomatoes	-0.017814	-0.070970	-0.034478	-0.021665	-0.045461	-0.009624
Waragi	-0.258802	0.012753	-0.152574	-0.160941	-0.176973	-0.035334
Yam	0.142130	-0.104841	0.006887	0.085868	0.032320	-0.026720

k	M 31-50	M 51+	log HSize	Constant
j				
Beans	0.011180	0.114473	0.391925	-0.765806
Beef	0.170483	0.171712	0.252517	-0.676754
Biscuits	0.170904	-0.127918	-0.075380	-0.409758
Bread	0.074366	0.078168	0.398146	-0.751295
Cabbages	0.030480	0.068790	0.232485	-0.513702
Cassava	0.098303	0.118365	0.148453	-0.711646
Chapati	0.093947	0.153264	0.162986	-0.412577
Cooking Oil	0.005265	0.052592	0.418256	-0.517449
Dodo	0.063169	0.162541	0.170590	-0.539539
Eggs	0.057990	0.005212	0.283776	-0.432881
Fish (dried)	0.080609	0.161793	0.234743	-0.503739
Fish (fresh)	0.207388	0.203726	0.031212	-0.528663
Ground Nuts	0.101304	0.148449	0.111036	-0.541421
Kabalagala	0.020680	-0.134739	0.407537	-0.414436
Maize	0.026396	0.083955	0.488098	-0.953142
Mangos	0.047207	0.141881	0.317855	-0.551921
Matoke	0.127467	0.155161	0.422577	-0.787167
Milk (fresh)	0.047731	0.127099	0.231194	-0.686888
Millet	0.100850	0.218176	0.316205	-0.756518
Onions	0.100514	0.087243	0.176612	-0.292793
Oranges	0.071078	0.195113	0.037732	-0.918389
Other Fruits	0.221923	0.211308	0.227456	-0.652330
Other Veg.	0.089325	0.144003	0.182118	-0.603278
Peas	-0.140722	-0.029455	0.556771	-0.739386
Pork	0.032198	0.146254	0.183903	-0.500035
Rice	0.002824	0.117627	0.338943	-0.747270
Salt	-0.023482	0.020827	0.408166	-0.662083
Sim Sim	0.259456	0.093475	0.203126	-0.601510
Soda	0.248710	0.183851	-0.096747	-0.245999
Sweet Bananas	0.055777	0.183890	0.261196	-0.768727
Sweet Potatoes	0.026218	0.057287	0.144537	-0.722471
Tea	0.134445	0.166611	0.169879	-0.519907
Tomatoes	0.038566	0.085942	0.304145	-0.436691
Waragi	0.110711	0.236917	0.539625	-0.318760
Yam	-0.078158	-0.045283	0.480181	-0.790806

4 (B) Nutritional Content of Different Foods

[8]:
```
food_nutrient = pd.read_excel("Uganda.xlsx", sheet_name = "FCT")
food_nutrient
```

[8]:
```
                     j  Energy  Protein  Fiber  Folate  Calcium  Carbohydrate  \
0              Avocado    1600     20.0   70.0     810      120          85.0
1          Beans (dry)    1700     98.0   60.0     500      580         325.0
2        Beans (fresh)    3470    214.0  160.0    5250     1130         626.0
3                 Beef    2510    182.0    0.0      60       70           0.0
4       Beef (roasted)    2910    264.0    0.0      70       90           0.0
..                 …       …        …      …       …        …             …
98            Tomatoes     180      9.0   10.0     150      100          39.0
99              Waragi    2630      0.0    0.0       0        0           0.0
100         Watermelon     300      6.1    4.0      30       70          75.5
101      Wheat (flour)    3640    103.0   30.0     260      150         763.0
102  Yams (arrowroot)    1180     15.0   40.0     230      170         279.0

     Iron  Niacin  Riboflavin  Thiamin  Vitamin A  Vitamin B-12  Vitamin B-6  \
0     6.0   17.38        1.30     0.67         70           0.0         2.57
1    30.0    7.00        1.20     3.40          0           0.0         2.15
2    51.0   11.74        2.12     7.13          0           0.0         4.74
3    19.0   31.50        1.60     0.90          0          28.9         3.80
4    27.0   37.20        2.20     0.90          0          24.7         3.40
..     …       …          …        …          …             …            …
98    3.0    5.94        0.19     0.37        420           0.0         0.80
99    0.0    0.00        0.00     0.00          0           0.0         0.00
100   2.4    1.78        0.21     0.33        280           0.0         0.45
101  12.0   12.50        0.40     1.20          0           0.0         0.44
102   5.0    5.52        0.32     1.12         70           0.0         2.93

     Vitamin C  Zinc
0          100   6.0
1           10   8.0
2           63  23.0
3            0  37.0
4            0  60.0
..         …     …
98         127   2.0
99           0   0.0
100         81   1.0
101          0   7.0
102        171   2.0

[103 rows x 16 columns]
```

5 (B) Nutritional Adequacy of Diet

[9]:
```
expenditure = pd.read_excel("Uganda.xlsx", sheet_name = "Expenditures␣
↪(2019-20)")
expenditure
```

[9]:
```
                                     i        t         m   Beans     Beef  \
0     00c9353d8ebe42faabf5919b81d7fae7  2019-20   Eastern  3600.0      NaN
1     062da72d5d3a457e9336b62c8bb9096d  2019-20   Eastern     NaN      NaN
2     0d0e29faff394154a69562b4527b48b8  2019-20   Eastern  1000.0   4500.0
3     0e03e253c35d4333a1ffad2df9d38850  2019-20   Eastern  2800.0      NaN
4                           1013000201  2019-20   Central     NaN      NaN
…                                  …       …        …       …        …
3004  bfdf0d66403440ceab439b1e1c47cdea  2019-20   Eastern  1200.0  10000.0
3005  c33f6cb57d9849949e08a7350dabb829  2019-20   Central     NaN      NaN
3006  d10a687889de469687377204195f3db0  2019-20   Western  2000.0      NaN
3007  d24fa50d02c041969a42102d8ebdadc9  2019-20   Eastern     NaN      NaN
3008  e07bc322c4884559b4b8ca75c945dd3e  2019-20  Northern  2600.0   3000.0

         Beer  Biscuits   Bongo  Bread  Butter, etc.  …  Sugarcane  \
0         NaN       NaN     NaN    NaN           NaN  …        NaN
1         NaN       NaN     NaN  500.0           NaN  …        NaN
2         NaN       NaN     NaN    NaN           NaN  …        NaN
3         NaN       NaN     NaN    NaN           NaN  …        NaN
4     17500.0       NaN     NaN    NaN           NaN  …        NaN
…         …         …       …      …            …  …        …
3004      NaN       NaN  2800.0    NaN           NaN  …        NaN
3005      NaN       NaN     NaN    NaN           NaN  …        NaN
3006      NaN       NaN     NaN    NaN           NaN  …        NaN
3007      NaN       NaN     NaN    NaN           NaN  …        NaN
3008      NaN       NaN     NaN    NaN           NaN  …        NaN

      Sweet Bananas  Sweet Potatoes    Tea  Tomatoes  Waragi  Water  \
0               NaN          4000.0  200.0    1000.0     NaN    NaN
1               NaN             NaN    NaN       NaN     NaN    NaN
2               NaN             NaN    NaN     800.0     NaN    NaN
3               NaN          5000.0  200.0     500.0     NaN    NaN
4            2000.0             NaN  400.0    2100.0     NaN    NaN
…               …             …      …        …       …
3004            NaN          6000.0  100.0    1000.0     NaN    NaN
3005            NaN          2000.0  200.0    1000.0     NaN    NaN
3006            NaN          2000.0    NaN    1000.0     NaN    NaN
3007            NaN         30000.0    NaN    1200.0     NaN    NaN
3008            NaN             NaN    NaN     500.0     NaN    NaN

      Wheat (flour)     Yam  Yogurt
0               NaN  3000.0     NaN
```

```
1               NaN     NaN     NaN
2               NaN     NaN     NaN
3               NaN     NaN     NaN
4               NaN     NaN     NaN
…                 …       …       …
3004            NaN  1000.0     NaN
3005            NaN     NaN     NaN
3006            NaN     NaN     NaN
3007            NaN     NaN     NaN
3008            NaN     NaN     NaN

[3009 rows x 77 columns]
```

[10]:
```python
price = pd.read_excel("Uganda.xlsx", sheet_name = "Prices")
price = price[price["t"] == "2019-20"]
price
```

[10]:
```
          t         m        Beans   Beef         Beer  Biscuits   Bongo  \
28  2019-20   Central  2500.000000  12000  6000.000000    5000.0  1000.0
29  2019-20   Eastern  2275.000000  10000  6785.714286    2000.0  1000.0
30  2019-20  Northern  8833.333333  10000  6500.000000    2000.0  1000.0
31  2019-20   Western  2200.000000  10000  5000.000000    2000.0  1250.0

     Bread  Butter, etc.     Cabbages  …    Sugarcane  Sweet Bananas  \
28  4500.0       10000.0  2683.028286  …  1679.389313    1169.607843
29  4500.0       10000.0  1679.389313  …  1428.571429    1225.000000
30  5000.0           NaN  2683.028286  …   714.285714    1398.039216
31  4500.0        6250.0  2351.145038  …  1341.514143    1720.588235

    Sweet Potatoes           Tea    Tomatoes  Waragi   Water  Wheat (flour)  \
28     1550.000000  13000.000000  671.755725  8000.0  1000.0         2875.0
29      794.444444  10000.000000  625.000000  6000.0  1600.0         2500.0
30     1000.000000  10000.000000  625.000000  4000.0  2000.0         3000.0
31      735.294118  11666.666667  750.000000  6000.0  2000.0         3200.0

            Yam  Yogurt
28  3358.778626  5000.0
29  3465.103599  7200.0
30  5366.056572  7000.0
31  5038.167939  8600.0

[4 rows x 76 columns]
```

[11]:
```python
foods = ['Beans', 'Beef', 'Beer', 'Biscuits', 'Bongo', 'Bread',
       'Butter, etc.', 'Cabbages', 'Cake', 'Cassava', 'Cassava (flour)',
       'Chapati', 'Cheese', 'Chicken', 'Cigarettes', 'Coffee', 'Cooking Oil',
       'Cornflakes', 'Dodo', 'Donut', 'Eggs', 'Fish (dried)', 'Fish (fresh)',
```

```
       'Garlic', 'Ghee', 'Ginger', 'Goat', 'Ground Nuts', 'Honey', 'Ice Cream',
       'Infant Formula', 'Irish Potatoes', 'Jackfruit', 'Jam/Marmalade',
       'Kabalagala', 'Macaroni/Spaghetti', 'Maize', 'Mangos', 'Matoke',
       'Milk (fresh)', 'Milk (powdered)', 'Millet', 'Onions', 'Oranges',
       'Other Alcohol', 'Other Drinks', 'Other Fruits', 'Other Juice',
       'Other Meat', 'Other Spices', 'Other Tobacco', 'Other Veg.',
       'Passion Fruits', 'Peas', 'Plantains', 'Pork', 'Rice', 'Salt', 'Samosa',
       'Sim Sim', 'Soda', 'Sorghum', 'Soybean', 'Sugar', 'Sugarcane',
       'Sweet Bananas', 'Sweet Potatoes', 'Tea', 'Tomatoes', 'Waragi', 'Water',
       'Wheat (flour)', 'Yam', 'Yogurt']
expenditure_and_price = expenditure.merge(price, how = "left", on = "m")
expenditure_and_price
```

[11]:

```
                                     i      t_x         m  Beans_x   Beef_x  \
0     00c9353d8ebe42faabf5919b81d7fae7  2019-20   Eastern   3600.0      NaN
1     062da72d5d3a457e9336b62c8bb9096d  2019-20   Eastern      NaN      NaN
2     0d0e29faff394154a69562b4527b48b8  2019-20   Eastern   1000.0   4500.0
3     0e03e253c35d4333a1ffad2df9d38850  2019-20   Eastern   2800.0      NaN
4                           1013000201  2019-20   Central      NaN      NaN
…                                    …        …         …        …        …
3004  bfdf0d66403440ceab439b1e1c47cdea  2019-20   Eastern   1200.0  10000.0
3005  c33f6cb57d9849949e08a7350dabb829  2019-20   Central      NaN      NaN
3006  d10a687889de469687377204195f3db0  2019-20   Western   2000.0      NaN
3007  d24fa50d02c041969a42102d8ebdadc9  2019-20   Eastern      NaN      NaN
3008  e07bc322c4884559b4b8ca75c945dd3e  2019-20  Northern   2600.0   3000.0

      Beer_x  Biscuits_x  Bongo_x  Bread_x  Butter, etc._x  …  Sugarcane_y  \
0        NaN         NaN      NaN      NaN             NaN  …  1428.571429
1        NaN         NaN      NaN    500.0             NaN  …  1428.571429
2        NaN         NaN      NaN      NaN             NaN  …  1428.571429
3        NaN         NaN      NaN      NaN             NaN  …  1428.571429
4    17500.0         NaN      NaN      NaN             NaN  …  1679.389313
…          …           …        …        …               …  …            …
3004     NaN         NaN   2800.0      NaN             NaN  …  1428.571429
3005     NaN         NaN      NaN      NaN             NaN  …  1679.389313
3006     NaN         NaN      NaN      NaN             NaN  …  1341.514143
3007     NaN         NaN      NaN      NaN             NaN  …  1428.571429
3008     NaN         NaN      NaN      NaN             NaN  …   714.285714

      Sweet Bananas_y  Sweet Potatoes_y         Tea_y  Tomatoes_y  Waragi_y  \
0         1225.000000        794.444444  10000.000000  625.000000    6000.0
1         1225.000000        794.444444  10000.000000  625.000000    6000.0
2         1225.000000        794.444444  10000.000000  625.000000    6000.0
3         1225.000000        794.444444  10000.000000  625.000000    6000.0
4         1169.607843       1550.000000  13000.000000  671.755725    8000.0
…                   …                 …             …           …         …
3004      1225.000000        794.444444  10000.000000  625.000000    6000.0
```

```
3005      1169.607843       1550.000000  13000.000000  671.755725     8000.0
3006      1720.588235        735.294118  11666.666667  750.000000     6000.0
3007      1225.000000        794.444444  10000.000000  625.000000     6000.0
3008      1398.039216       1000.000000  10000.000000  625.000000     4000.0

      Water_y  Wheat (flour)_y          Yam_y  Yogurt_y
0      1600.0           2500.0  3465.103599    7200.0
1      1600.0           2500.0  3465.103599    7200.0
2      1600.0           2500.0  3465.103599    7200.0
3      1600.0           2500.0  3465.103599    7200.0
4      1000.0           2875.0  3358.778626    5000.0
…        …              …          …        …
3004   1600.0           2500.0  3465.103599    7200.0
3005   1000.0           2875.0  3358.778626    5000.0
3006   2000.0           3200.0  5038.167939    8600.0
3007   1600.0           2500.0  3465.103599    7200.0
3008   2000.0           3000.0  5366.056572    7000.0

[3009 rows x 152 columns]
```

[12]:
```python
for food in foods:
    exp = str(food) + "_x"
    price = str(food) + "_y"
    expenditure_and_price[food] = expenditure_and_price[exp]/
 ↪expenditure_and_price[price]
household_consumption = expenditure_and_price[foods].fillna(0)
household_consumption
```

[12]:
```
         Beans  Beef       Beer  Biscuits  Bongo     Bread  Butter, etc.  \
0     1.582418  0.00  0.000000       0.0    0.0  0.000000           0.0
1     0.000000  0.00  0.000000       0.0    0.0  0.111111           0.0
2     0.439560  0.45  0.000000       0.0    0.0  0.000000           0.0
3     1.230769  0.00  0.000000       0.0    0.0  0.000000           0.0
4     0.000000  0.00  2.916667       0.0    0.0  0.000000           0.0
…          …    …         …         …      …         …            …
3004  0.527473  1.00  0.000000       0.0    2.8  0.000000           0.0
3005  0.000000  0.00  0.000000       0.0    0.0  0.000000           0.0
3006  0.909091  0.00  0.000000       0.0    0.0  0.000000           0.0
3007  0.000000  0.00  0.000000       0.0    0.0  0.000000           0.0
3008  0.294340  0.30  0.000000       0.0    0.0  0.000000           0.0

      Cabbages  Cake   Cassava  …  Sugarcane  Sweet Bananas  Sweet Potatoes  \
0     0.833636   0.0  4.857143  …        0.0       0.000000        5.034965
1     0.000000   0.0  0.000000  …        0.0       0.000000        0.000000
2     0.000000   0.0  3.238095  …        0.0       0.000000        0.000000
3     0.000000   0.0  2.590476  …        0.0       0.000000        6.293706
4     0.000000   0.0  2.857143  …        0.0       1.709975        0.000000
```

```
…         …    …         …   …          …             …              …
3004  0.297727   0.0  2.590476  …         0.0       0.000000        7.552448
3005  0.000000   0.0  1.785714  …         0.0       0.000000        1.290323
3006  0.425325   0.0  0.000000  …         0.0       0.000000        2.720000
3007  0.000000   0.0  0.000000  …         0.0       0.000000       37.762238
3008  0.000000   0.0  0.000000  …         0.0       0.000000        0.000000

           Tea  Tomatoes  Waragi  Water  Wheat (flour)       Yam  Yogurt
0     0.020000  1.600000     0.0    0.0            0.0  0.865775     0.0
1     0.000000  0.000000     0.0    0.0            0.0  0.000000     0.0
2     0.000000  1.280000     0.0    0.0            0.0  0.000000     0.0
3     0.020000  0.800000     0.0    0.0            0.0  0.000000     0.0
4     0.030769  3.126136     0.0    0.0            0.0  0.000000     0.0
…          …         …      …      …              …         …       …
3004  0.010000  1.600000     0.0    0.0            0.0  0.288592     0.0
3005  0.015385  1.488636     0.0    0.0            0.0  0.000000     0.0
3006  0.000000  1.333333     0.0    0.0            0.0  0.000000     0.0
3007  0.000000  1.920000     0.0    0.0            0.0  0.000000     0.0
3008  0.000000  0.800000     0.0    0.0            0.0  0.000000     0.0

[3009 rows x 74 columns]
```

[13]:
```python
#household dempgrahics
household = pd.read_excel("Uganda.xlsx", sheet_name = "HH Characteristics")
household_19_20 = household[household["t"] == "2019-20"]
household_19_20
```

[13]:
```
                                      i        t         m  F 00-03  F 04-08  \
1      00c9353d8ebe42faabf5919b81d7fae7  2019-20   Eastern      1.0      0.0
6      062da72d5d3a457e9336b62c8bb9096d  2019-20   Eastern      0.0      0.0
8      0d0e29faff394154a69562b4527b48b8  2019-20   Eastern      1.0      0.0
10     0e03e253c35d4333a1ffad2df9d38850  2019-20   Eastern      1.0      1.0
18                           1013000201  2019-20   Central      0.0      0.0
…                                     …        …         …        …        …
24349  c33f6cb57d9849949e08a7350dabb829  2019-20   Central      0.0      0.0
24352  d10a687889de469687377204195f3db0  2019-20   Western      0.0      0.0
24354  d24fa50d02c041969a42102d8ebdadc9  2019-20   Eastern      0.0      1.0
24357  e07bc322c4884559b4b8ca75c945dd3e  2019-20  Northern      1.0      1.0
24361  ef69f1cfdaf44c1e81a81bf21c2981f4  2019-20   Central      0.0      0.0

       F 09-13  F 14-18  F 19-30  F 31-50  F 51+  M 00-03  M 04-08  M 09-13  \
1          0.0      3.0      1.0      0.0    1.0      0.0      0.0      1.0
6          0.0      0.0      0.0      0.0    0.0      0.0      0.0      0.0
8          0.0      0.0      1.0      0.0    0.0      1.0      0.0      0.0
10         1.0      0.0      1.0      0.0    0.0      0.0      0.0      0.0
18         0.0      0.0      0.0      0.0    1.0      0.0      0.0      0.0
…            …        …        …        …      …        …        …        …
```

```
24349       0.0      0.0      0.0      0.0    1.0      0.0      0.0      0.0
24352       0.0      0.0      0.0      0.0    1.0      0.0      0.0      0.0
24354       1.0      1.0      0.0      1.0    0.0      1.0      0.0      1.0
24357       1.0      0.0      0.0      1.0    0.0      1.0      1.0      1.0
24361       0.0      0.0      0.0      0.0    0.0      0.0      0.0      0.0

       M 14-18  M 19-30  M 31-50  M 51+  log HSize
1          0.0      0.0      0.0    0.0   1.945910
6          0.0      0.0      1.0    0.0   0.000000
8          0.0      0.0      1.0    0.0   1.386294
10         0.0      0.0      1.0    0.0   1.609438
18         0.0      0.0      0.0    0.0   0.000000
…        …      …      …      …      …
24349      0.0      0.0      0.0    0.0   0.000000
24352      1.0      0.0      1.0    0.0   1.098612
24354      0.0      1.0      0.0    1.0   2.079442
24357      0.0      0.0      0.0    0.0   1.945910
24361      0.0      0.0      1.0    0.0   0.000000

[3076 rows x 18 columns]
```

[14]:
```
household_consumption['i'] = expenditure["i"]
household_consumption = household_consumption.merge(household_19_20, how =
↪"left", on = "i")
household_consumption
```

[14]:
```
         Beans  Beef      Beer  Biscuits  Bongo     Bread  Butter, etc.  \
0     1.582418  0.00  0.000000       0.0    0.0  0.000000           0.0
1     0.000000  0.00  0.000000       0.0    0.0  0.111111           0.0
2     0.439560  0.45  0.000000       0.0    0.0  0.000000           0.0
3     1.230769  0.00  0.000000       0.0    0.0  0.000000           0.0
4     0.000000  0.00  2.916667       0.0    0.0  0.000000           0.0
…          …    …         …       …      …       …           …
3004  0.527473  1.00  0.000000       0.0    2.8  0.000000           0.0
3005  0.000000  0.00  0.000000       0.0    0.0  0.000000           0.0
3006  0.909091  0.00  0.000000       0.0    0.0  0.000000           0.0
3007  0.000000  0.00  0.000000       0.0    0.0  0.000000           0.0
3008  0.294340  0.30  0.000000       0.0    0.0  0.000000           0.0

      Cabbages  Cake   Cassava  …  F 31-50  F 51+  M 00-03  M 04-08  \
0     0.833636   0.0  4.857143  …      0.0    1.0      0.0      0.0
1     0.000000   0.0  0.000000  …      0.0    0.0      0.0      0.0
2     0.000000   0.0  3.238095  …      0.0    0.0      1.0      0.0
3     0.000000   0.0  2.590476  …      0.0    0.0      0.0      0.0
4     0.000000   0.0  2.857143  …      NaN    NaN      NaN      NaN
…          …    …        …   …      …      …       …       …
3004  0.297727   0.0  2.590476  …      1.0    0.0      3.0      0.0
```

```
3005  0.000000   0.0  1.785714  …      0.0     1.0       0.0       0.0
3006  0.425325   0.0  0.000000  …      0.0     1.0       0.0       0.0
3007  0.000000   0.0  0.000000  …      1.0     0.0       1.0       0.0
3008  0.000000   0.0  0.000000  …      1.0     0.0       1.0       1.0

      M 09-13  M 14-18  M 19-30  M 31-50  M 51+  log HSize
0         1.0      0.0      0.0      0.0    0.0   1.945910
1         0.0      0.0      0.0      1.0    0.0   0.000000
2         0.0      0.0      0.0      1.0    0.0   1.386294
3         0.0      0.0      0.0      1.0    0.0   1.609438
4         NaN      NaN      NaN      NaN    NaN        NaN
…       …      …      …      …     …         …
3004      1.0      1.0      0.0      0.0    1.0   2.197225
3005      0.0      0.0      0.0      0.0    0.0   0.000000
3006      0.0      1.0      0.0      1.0    0.0   1.098612
3007      1.0      0.0      1.0      0.0    1.0   2.079442
3008      1.0      0.0      0.0      0.0    0.0   1.945910

[3009 rows x 92 columns]
```

[15]:
```python
RDI = pd.read_excel("Uganda.xlsx", sheet_name = "RDI")
RDI
```

[15]:
```
               n  F 00-03  M 00-03  F 04-08  M 04-08  F 09-13  M 09-13  \
0         Energy   1000.0   1000.0   1200.0   1400.0   1600.0   1800.0
1        Protein     13.0     13.0     19.0     19.0     34.0     34.0
2          Fiber     14.0     14.0     16.8     19.6     22.4     25.2
3         Folate    150.0    150.0    200.0    200.0    300.0    300.0
4        Calcium    700.0    700.0   1000.0   1000.0   1300.0   1300.0
5   Carbohydrate    130.0    130.0    130.0    130.0    130.0    130.0
6           Iron      7.0      7.0     10.0     10.0      8.0      8.0
7      Magnesium     80.0     80.0    130.0    130.0    240.0    240.0
8         Niacin      6.0      6.0      8.0      8.0     12.0     12.0
9     Phosphorus    460.0    460.0    500.0    500.0   1250.0   1250.0
10     Potassium   3000.0   3000.0   3800.0   3800.0   4500.0   4500.0
11    Riboflavin      0.5      0.5      0.6      0.6      0.9      0.9
12       Thiamin      0.5      0.5      0.6      0.6      0.9      0.9
13     Vitamin A    300.0    300.0    400.0    400.0    600.0    600.0
14  Vitamin B-12      0.9      0.9      1.2      1.2      1.8      1.8
15   Vitamin B-6      0.5      0.5      0.6      0.6      1.0      1.0
16     Vitamin C     15.0     15.0     25.0     25.0     45.0     45.0
17     Vitamin E      6.0      6.0      7.0      7.0     11.0     11.0
18     Vitamin K     30.0     30.0     55.0     55.0     60.0     60.0
19          Zinc      3.0      3.0      5.0      5.0      8.0      8.0

    F 14-18  M 14-18  F 19-30  M 19-30  F 31-50  M 31-50   F 51+   M 51+
0    1800.0   2200.0   2000.0   2400.0   1800.0   2200.0  1600.0  2000.0
```

```
1     46.0    52.0    46.0    56.0    46.0    56.0    46.0    56.0
2     25.2    30.8    28.0    33.6    25.2    30.8    22.4    28.0
3    400.0   400.0   400.0   400.0   400.0   400.0   400.0   400.0
4   1300.0  1300.0  1000.0  1000.0  1000.0  1000.0  1200.0  1000.0
5    130.0   130.0   130.0   130.0   130.0   130.0   130.0   130.0
6     15.0    11.0    18.0     8.0    18.0     8.0     8.0     8.0
7    360.0   410.0   310.0   400.0   320.0   420.0   320.0   420.0
8     14.0    16.0    14.0    16.0    14.0    16.0    14.0    16.0
9   1250.0  1250.0   700.0   700.0   700.0   700.0   700.0   700.0
10  4700.0  4700.0  4700.0  4700.0  4700.0  4700.0  4700.0  4700.0
11     1.0     1.3     1.1     1.3     1.1     1.3     1.1     1.3
12     1.0     1.2     1.1     1.2     1.1     1.2     1.1     1.2
13   700.0   900.0   700.0   900.0   700.0   900.0   700.0   900.0
14     2.4     2.4     2.4     2.4     2.4     2.4     2.4     2.4
15     1.2     1.3     1.3     1.3     1.3     1.3     1.5     1.7
16    65.0    75.0    75.0    90.0    75.0    90.0    75.0    90.0
17    15.0    15.0    15.0    15.0    15.0    15.0    15.0    15.0
18    75.0    75.0    90.0   120.0    90.0   120.0    90.0   120.0
19     9.0    11.0     8.0    11.0     8.0    11.0     8.0    11.0
```

[16]:
```
x = household_consumption[['F 00-03', 'F 04-08', 'F 09-13', 'F 14-18', 'F␣
↪19-30',
       'F 31-50', 'F 51+', 'M 00-03', 'M 04-08', 'M 09-13', 'M 14-18',
       'M 19-30', 'M 31-50', 'M 51+']]
x
```

[16]:
```
      F 00-03  F 04-08  F 09-13  F 14-18  F 19-30  F 31-50  F 51+  M 00-03  \
0         1.0      0.0      0.0      3.0      1.0      0.0    1.0      0.0
1         0.0      0.0      0.0      0.0      0.0      0.0    0.0      0.0
2         1.0      0.0      0.0      0.0      1.0      0.0    0.0      1.0
3         1.0      1.0      1.0      0.0      1.0      0.0    0.0      0.0
4         NaN      NaN      NaN      NaN      NaN      NaN    NaN      NaN
…         …        …        …        …        …        …      …        …
3004      0.0      1.0      1.0      0.0      0.0      1.0    0.0      3.0
3005      0.0      0.0      0.0      0.0      0.0      0.0    1.0      0.0
3006      0.0      0.0      0.0      0.0      0.0      0.0    1.0      0.0
3007      0.0      1.0      1.0      1.0      0.0      1.0    0.0      1.0
3008      1.0      1.0      1.0      0.0      0.0      1.0    0.0      1.0

      M 04-08  M 09-13  M 14-18  M 19-30  M 31-50  M 51+
0         0.0      1.0      0.0      0.0      0.0    0.0
1         0.0      0.0      0.0      0.0      1.0    0.0
2         0.0      0.0      0.0      0.0      1.0    0.0
3         0.0      0.0      0.0      0.0      1.0    0.0
4         NaN      NaN      NaN      NaN      NaN    NaN
…         …        …        …        …        …      …
3004      0.0      1.0      1.0      0.0      0.0    1.0
```

```
3005       0.0       0.0       0.0       0.0       0.0     0.0
3006       0.0       0.0       1.0       0.0       1.0     0.0
3007       0.0       1.0       0.0       1.0       0.0     1.0
3008       1.0       1.0       0.0       0.0       0.0     0.0

[3009 rows x 14 columns]
```

[17]:
```python
y = RDI[['F 00-03', 'F 04-08', 'F 09-13', 'F 14-18', 'F 19-30',
       'F 31-50', 'F 51+', 'M 00-03', 'M 04-08', 'M 09-13', 'M 14-18',
       'M 19-30', 'M 31-50', 'M 51+']].transpose()
y
```

[17]:
```
              0     1     2      3       4      5     6      7     8       9  \
F 00-03  1000.0  13.0  14.0  150.0   700.0  130.0   7.0   80.0   6.0   460.0
F 04-08  1200.0  19.0  16.8  200.0  1000.0  130.0  10.0  130.0   8.0   500.0
F 09-13  1600.0  34.0  22.4  300.0  1300.0  130.0   8.0  240.0  12.0  1250.0
F 14-18  1800.0  46.0  25.2  400.0  1300.0  130.0  15.0  360.0  14.0  1250.0
F 19-30  2000.0  46.0  28.0  400.0  1000.0  130.0  18.0  310.0  14.0   700.0
F 31-50  1800.0  46.0  25.2  400.0  1000.0  130.0  18.0  320.0  14.0   700.0
F 51+    1600.0  46.0  22.4  400.0  1200.0  130.0   8.0  320.0  14.0   700.0
M 00-03  1000.0  13.0  14.0  150.0   700.0  130.0   7.0   80.0   6.0   460.0
M 04-08  1400.0  19.0  19.6  200.0  1000.0  130.0  10.0  130.0   8.0   500.0
M 09-13  1800.0  34.0  25.2  300.0  1300.0  130.0   8.0  240.0  12.0  1250.0
M 14-18  2200.0  52.0  30.8  400.0  1300.0  130.0  11.0  410.0  16.0  1250.0
M 19-30  2400.0  56.0  33.6  400.0  1000.0  130.0   8.0  400.0  16.0   700.0
M 31-50  2200.0  56.0  30.8  400.0  1000.0  130.0   8.0  420.0  16.0   700.0
M 51+    2000.0  56.0  28.0  400.0  1000.0  130.0   8.0  420.0  16.0   700.0

             10   11   12     13   14   15    16    17     18    19
F 00-03  3000.0  0.5  0.5  300.0  0.9  0.5  15.0   6.0   30.0   3.0
F 04-08  3800.0  0.6  0.6  400.0  1.2  0.6  25.0   7.0   55.0   5.0
F 09-13  4500.0  0.9  0.9  600.0  1.8  1.0  45.0  11.0   60.0   8.0
F 14-18  4700.0  1.0  1.0  700.0  2.4  1.2  65.0  15.0   75.0   9.0
F 19-30  4700.0  1.1  1.1  700.0  2.4  1.3  75.0  15.0   90.0   8.0
F 31-50  4700.0  1.1  1.1  700.0  2.4  1.3  75.0  15.0   90.0   8.0
F 51+    4700.0  1.1  1.1  700.0  2.4  1.5  75.0  15.0   90.0   8.0
M 00-03  3000.0  0.5  0.5  300.0  0.9  0.5  15.0   6.0   30.0   3.0
M 04-08  3800.0  0.6  0.6  400.0  1.2  0.6  25.0   7.0   55.0   5.0
M 09-13  4500.0  0.9  0.9  600.0  1.8  1.0  45.0  11.0   60.0   8.0
M 14-18  4700.0  1.3  1.2  900.0  2.4  1.3  75.0  15.0   75.0  11.0
M 19-30  4700.0  1.3  1.2  900.0  2.4  1.3  90.0  15.0  120.0  11.0
M 31-50  4700.0  1.3  1.2  900.0  2.4  1.3  90.0  15.0  120.0  11.0
M 51+    4700.0  1.3  1.2  900.0  2.4  1.7  90.0  15.0  120.0  11.0
```

[18]:
```python
required_nutrients_household = x@y
required_nutrients_household.columns = RDI["n"]
required_nutrients_household
```

[18]:
```
n       Energy  Protein  Fiber  Folate  Calcium  Carbohydrate  Iron  Magnesium  \
0      11800.0    277.0  165.2  2450.0   8100.0         910.0  86.0     2030.0
1       2200.0     56.0   30.8   400.0   1000.0         130.0   8.0      420.0
2       6200.0    128.0   86.8  1100.0   3400.0         520.0  40.0      890.0
3       8000.0    168.0  112.0  1450.0   5000.0         650.0  51.0     1180.0
4          NaN      NaN    NaN     NaN      NaN           NaN   NaN        NaN
…          …      …    …      …      …        …   …          …
3004   13600.0    280.0  190.4  2450.0   9000.0        1170.0  84.0     2000.0
3005    1600.0     46.0   22.4   400.0   1200.0         130.0   8.0      320.0
3006    6000.0    154.0   84.0  1200.0   3500.0         390.0  27.0     1150.0
3007   13600.0    304.0  190.4  2550.0   8600.0        1040.0  82.0     2190.0
3008    9800.0    178.0  137.2  1700.0   7000.0         910.0  68.0     1220.0

n      Niacin  Phosphorus  Potassium  Riboflavin  Thiamin  Vitamin A  \
0        88.0      6860.0    31000.0         6.6      6.6     4400.0
1        16.0       700.0     4700.0         1.3      1.2      900.0
2        42.0      2320.0    15400.0         3.4      3.3     2200.0
3        56.0      3610.0    20700.0         4.4      4.3     2900.0
4         NaN         NaN        NaN         NaN      NaN        NaN
…         …           …          …           …        …
3004     96.0      7030.0    35900.0         7.6      7.4     5000.0
3005     14.0       700.0     4700.0         1.1      1.1      700.0
3006     46.0      2650.0    14100.0         3.7      3.5     2500.0
3007     98.0      6810.0    34600.0         7.6      7.4     5100.0
3008     66.0      5120.0    27300.0         5.1      5.1     3300.0

n      Vitamin B-12  Vitamin B-6  Vitamin C  Vitamin E  Vitamin K  Zinc
0              14.7          7.9      405.0       92.0      495.0  54.0
1               2.4          1.3       90.0       15.0      120.0  11.0
2               6.6          3.6      195.0       42.0      270.0  25.0
3               8.7          4.7      250.0       54.0      355.0  35.0
4               NaN          NaN        NaN        NaN        NaN   NaN
…               …            …        …          …         …    …
3004           14.7          8.4      400.0       92.0      550.0  60.0
3005            2.4          1.5       75.0       15.0       90.0   8.0
3006            7.2          4.1      240.0       45.0      285.0  30.0
3007           15.3          8.6      450.0       95.0      610.0  63.0
3008           10.2          5.5      245.0       63.0      380.0  40.0

[3009 rows x 20 columns]
```

[19]:
```
food_consumed = ['Beans', 'Beef', 'Beer', 'Biscuits', 'Bongo',
       'Bread', 'Butter, etc.', 'Cabbages', 'Cake', 'Cassava',
       'Cassava (flour)', 'Chapati', 'Cheese', 'Chicken', 'Cigarettes',
       'Coffee', 'Cooking Oil', 'Cornflakes', 'Dodo', 'Donut', 'Eggs',
       'Fish (dried)', 'Fish (fresh)', 'Garlic', 'Ghee', 'Ginger', 'Goat',
       'Ground Nuts', 'Honey', 'Ice Cream', 'Infant Formula', 'Irish Potatoes',
```

```
'Jackfruit', 'Jam/Marmalade', 'Kabalagala', 'Macaroni/Spaghetti',
'Maize', 'Mangos', 'Matoke', 'Milk (fresh)', 'Milk (powdered)',
'Millet', 'Onions', 'Oranges', 'Other Alcohol', 'Other Drinks',
'Other Fruits', 'Other Juice', 'Other Meat', 'Other Spices',
'Other Tobacco', 'Other Veg.', 'Passion Fruits', 'Peas', 'Plantains',
'Pork', 'Rice', 'Salt', 'Samosa', 'Sim Sim', 'Soda', 'Sorghum',
'Soybean', 'Sugar', 'Sugarcane', 'Sweet Bananas', 'Sweet Potatoes',
'Tea', 'Tomatoes', 'Waragi', 'Water', 'Wheat (flour)', 'Yam', 'Yogurt']
```

[20]:
```
food_nutrient = food_nutrient[food_nutrient['j'].isin(food_consumed)]
x_2 = household_consumption[food_nutrient['j']]
x_2 = x_2.fillna(0)
x_2
```

[20]:
```
      Beef  Biscuits  Bongo     Bread  Cabbages  Cassava (flour)    Chapati  \
0     0.00       0.0    0.0  0.000000  0.833636              0.0   1.714286
1     0.00       0.0    0.0  0.111111  0.000000              0.0   0.571429
2     0.45       0.0    0.0  0.000000  0.000000              0.0   0.000000
3     0.00       0.0    0.0  0.000000  0.000000              0.0   0.857143
4     0.00       0.0    0.0  0.000000  0.000000              0.0  12.000000
…      …        …     …       …        …                …         …
3004  1.00       0.0    2.8  0.000000  0.297727              0.0   1.714286
3005  0.00       0.0    0.0  0.000000  0.000000              0.0   0.000000
3006  0.00       0.0    0.0  0.000000  0.425325              0.0   0.000000
3007  0.00       0.0    0.0  0.000000  0.000000              0.0   0.000000
3008  0.30       0.0    0.0  0.000000  0.000000              0.0   0.000000

      Cooking Oil  Dodo  Eggs  …      Pork  Sim Sim  Soda   Sorghum  Sugar  \
0        0.810811  3.00   0.0  …  0.000000      0.0  0.60  0.000000    0.0
1        0.000000  0.00   0.0  …  0.000000      0.0  0.60  0.000000    0.0
2        0.000000  0.00   0.0  …  0.000000      0.0  0.00  1.500000    0.0
3        0.486486  2.40   0.0  …  0.000000      0.0  0.00  0.000000    0.0
4        0.660000  0.00   0.0  …  0.833333      0.0  0.72  0.000000    0.0
…             …    …    …   …  …       …    …      …    …
3004     0.432432  0.75   0.0  …  0.000000      0.0  0.90  0.000000    0.0
3005     0.000000  0.90   0.0  …  0.000000      0.0  0.00  0.000000    0.0
3006     0.000000  0.00   0.0  …  0.000000      0.0  0.00  0.000000    0.0
3007     0.000000  2.70   0.0  …  0.000000      0.0  0.00  0.000000    0.0
3008     1.320000  8.00   0.0  …  0.000000      0.0  0.00  6.666667    0.0

      Sugarcane  Sweet Bananas  Tomatoes  Waragi  Wheat (flour)
0           0.0       0.000000  1.600000     0.0            0.0
1           0.0       0.000000  0.000000     0.0            0.0
2           0.0       0.000000  1.280000     0.0            0.0
3           0.0       0.000000  0.800000     0.0            0.0
4           0.0       1.709975  3.126136     0.0            0.0
…          …            …       …      …            …
```

```
3004         0.0       0.000000  1.600000     0.0            0.0
3005         0.0       0.000000  1.488636     0.0            0.0
3006         0.0       0.000000  1.333333     0.0            0.0
3007         0.0       0.000000  1.920000     0.0            0.0
3008         0.0       0.000000  0.800000     0.0            0.0

[3009 rows x 33 columns]
```

[21]:
```python
y_2 = food_nutrient.iloc[:,1:].set_index(food_nutrient['j'])
y_2
```

[21]:
```
                    Energy  Protein  Fiber  Folate  Calcium  Carbohydrate  \
j
Beef                  2510    182.0    0.0      60       70           0.0
Biscuits              4460     69.0   10.0    1030      430         741.0
Bongo                  640     33.0    0.0      50     1620          45.0
Bread                 2660     76.0   20.0    1110     1510         506.0
Cabbages               250     13.0   30.0     430      400          58.0
Cassava (flour)       3140     26.0   40.0     360      310         766.0
Chapati               2750     91.0   20.0     240      860         557.0
Cooking Oil           8840      0.0    0.0       0        0           0.0
Dodo                   230     25.0    0.0     850     2150          40.0
Eggs                  1430    126.0    0.0     470      530           8.0
Garlic                1490     64.0   20.0      30     1810         331.0
Ghee                  8760      3.0    0.0       0       40           0.0
Honey                 3040      3.0    0.0      20       60         824.0
Irish Potatoes         770     20.0   20.0     160      120         175.0
Kabalagala            2540     17.0   30.0     160      200         487.0
Macaroni/Spaghetti    3710    130.0   32.0     180      210         747.0
Mangos                 650      5.0   20.0     140      100         170.0
Milk (powdered)       4960    263.0    0.0     370     9120         384.0
Millet                3780    110.0   90.0     850       80         729.0
Onions                 400     11.0   20.0     190      230          93.0
Oranges                470      9.0   20.0     300      400         118.0
Passion Fruits         970     22.0  100.0     140      120         234.0
Peas                   900     30.0   50.0    1680     1260         188.0
Pork                  2000    195.0    0.0      50      190           0.0
Sim Sim               5730    177.0  118.0     970     9750         235.0
Soda                   480      0.0    0.0       0       50         123.0
Sorghum               3390    113.0   60.0     140      280         746.0
Sugar                 3870      0.0    0.0       0       10        1000.0
Sugarcane              540      6.0   31.0       0       80         130.0
Sweet Bananas          890     11.0   30.0     200       50         228.0
Tomatoes               180      9.0   10.0     150      100          39.0
Waragi                2630      0.0    0.0       0        0           0.0
Wheat (flour)         3640    103.0   30.0     260      150         763.0
```

j	Iron	Niacin	Riboflavin	Thiamin	Vitamin A \
Beef	19.0	31.50	1.60	0.90	0
Biscuits	28.0	34.70	3.26	3.50	0
Bongo	1.0	1.00	1.50	0.20	370
Bread	37.0	43.85	3.31	4.55	0
Cabbages	5.0	2.34	0.40	0.61	50
Cassava (flour)	19.0	14.00	0.50	3.10	70
Chapati	14.0	21.42	0.97	2.67	0
Cooking Oil	0.0	0.00	0.00	0.00	0
Dodo	23.0	6.58	1.58	0.27	1460
Eggs	18.0	0.70	4.78	0.69	1400
Garlic	17.0	7.00	1.10	2.00	0
Ghee	0.0	0.03	0.05	0.01	8400
Honey	4.0	1.21	0.38	0.00	0
Irish Potatoes	8.0	10.54	0.32	0.80	0
Kabalagala	11.0	8.60	0.38	1.26	1100
Macaroni/Spaghetti	13.0	17.00	0.60	0.90	0
Mangos	1.0	5.84	0.57	0.58	380
Milk (powdered)	5.0	6.46	12.05	2.83	2570
Millet	30.0	47.20	2.90	4.21	0
Onions	2.0	1.20	0.30	0.50	0
Oranges	1.0	2.82	0.40	0.87	110
Passion Fruits	16.0	15.00	1.30	0.00	640
Peas	11.0	14.50	1.45	1.10	410
Pork	8.0	44.90	2.50	8.90	20
Sim Sim	145.5	45.20	2.50	7.90	0
Soda	1.0	0.00	0.00	0.00	0
Sorghum	44.0	29.27	1.42	2.37	0
Sugar	0.0	0.00	0.19	0.00	0
Sugarcane	14.0	1.00	0.10	0.20	0
Sweet Bananas	3.0	6.65	0.73	0.31	30
Tomatoes	3.0	5.94	0.19	0.37	420
Waragi	0.0	0.00	0.00	0.00	0
Wheat (flour)	12.0	12.50	0.40	1.20	0

j	Vitamin B-12	Vitamin B-6	Vitamin C	Zinc
Beef	28.9	3.80	0	37.0
Biscuits	0.5	0.22	0	6.0
Bongo	4.4	0.36	0	6.0
Bread	0.0	0.84	0	7.0
Cabbages	0.0	1.24	366	2.0
Cassava (flour)	0.0	7.00	720	7.0
Chapati	0.0	0.34	0	8.0
Cooking Oil	0.0	0.00	0	0.0
Dodo	0.0	1.92	433	9.0

```
Eggs                         12.9         1.43          0   11.0
Garlic                        0.0        12.35        312   12.0
Ghee                          0.1         0.01          0    0.0
Honey                         0.0         0.24          5    2.0
Irish Potatoes                0.0         2.95        197    3.0
Kabalagala                    0.0         4.57        320    4.0
Macaroni/Spaghetti            0.0         1.42          0   14.0
Mangos                        0.0         1.34        277    0.0
Milk (powdered)              32.5         3.02         86   33.0
Millet                        0.0         3.84          0   17.0
Onions                        0.0         1.20         74    2.0
Oranges                       0.0         0.60        532    1.0
Passion Fruits                0.0         1.00        300    1.0
Peas                          0.0         0.67         25   10.0
Pork                          6.3         4.60          6   19.0
Sim Sim                       0.0         7.90          0   77.5
Soda                          0.0         0.00          0    1.0
Sorghum                       0.0         1.50          0   16.0
Sugar                         0.0         0.00          0    0.0
Sugarcane                     0.0         0.00         30    0.0
Sweet Bananas                 0.0         3.67         87    2.0
Tomatoes                      0.0         0.80        127    2.0
Waragi                        0.0         0.00          0    0.0
Wheat (flour)                 0.0         0.44          0    7.0
```

[22]:
```
consumed_nutrients = x_2@y_2
consumed_nutrients
```

[22]:
```
            Energy      Protein        Fiber       Folate       Calcium  \
0     21027.690944   323.451558   202.437662  4666.320779   9877.025974
1      2154.984127    60.444444    13.650794   260.476190    689.206349
2      6604.900000   267.320000   110.800000   505.000000    671.500000
3     14730.826255   197.071429   116.857143  2879.857143   6617.571429
4     43099.248832  1306.064951   330.960609  3812.382092  11009.045712
…              …            …            …            …            …
3004  23103.063092   556.391883   232.931818  3350.094156  10280.305195
3005    594.954545    39.197727    20.886364  1045.295455   2152.863636
3006   8167.988312   172.347792   174.378788  1642.746753    698.234632
3007   1046.600000    86.980000    23.200000  2621.000000   6043.000000
3008  38301.800000  1030.173333   452.800000  8216.933333  19474.866667

      Carbohydrate        Iron      Niacin  Riboflavin    Thiamin  \
0      2794.193766  133.853896   96.268995    8.967455  11.717661
1       448.307937   12.711111   17.112222    0.922063   2.031270
2      1206.120000   79.190000   66.163200    3.213200   4.633600
3      2023.957143  101.628571   63.835429    5.951143   6.982571
4      7323.413585  190.735000  324.901249   15.691581  41.353429
```

```
…            …           …           …          …          …
3004   3201.853896  106.467208  122.487110  11.876662  14.293256
3005    121.956818   25.765909   15.124500   1.794841   0.943795
3006   1535.940260   55.298052   75.744731   4.429249   7.395081
3007    201.480000   68.260000   29.410800   4.690800   1.539400
3008   5656.053333  488.313333  272.087333  23.842667  19.774000

         Vitamin A  Vitamin B-12  Vitamin B-6    Vitamin C        Zinc
0      8142.253247         0.000    22.207995  3731.168052   59.467273
1         0.000000         0.000     0.287619     0.000000    5.949206
2       537.600000        13.005     5.464000   192.160000   44.010000
3      6982.857143         0.000    18.956571  2077.285714   42.085714
4      1380.943185         5.250    17.193850   581.867130  123.065556
…              …           …           …           …          …
3004   6400.743506        41.220    24.417182  3708.803896   95.188312
3005   1939.227273         0.000     3.278909   600.956818   11.677273
3006   2090.409091         0.003    11.785226   712.916450   29.088745
3007   4748.400000         0.000     6.960000  1427.740000   28.540000
3008  12624.000000         8.670    30.052000  4056.160000  192.646667

[3009 rows x 15 columns]
```

[23]:
```
required_nutrients_household = required_nutrients_household[consumed_nutrients.
  ↪columns].fillna(0)
required_nutrients_household*7
```

[23]:
```
n      Energy  Protein   Fiber  Folate  Calcium  Carbohydrate   Iron  Niacin  \
0     82600.0   1939.0  1156.4  17150.0  56700.0        6370.0  602.0   616.0
1     15400.0    392.0   215.6   2800.0   7000.0         910.0   56.0   112.0
2     43400.0    896.0   607.6   7700.0  23800.0        3640.0  280.0   294.0
3     56000.0   1176.0   784.0  10150.0  35000.0        4550.0  357.0   392.0
4         0.0      0.0     0.0      0.0      0.0           0.0    0.0     0.0
…        …       …       …       …       …        …      …      …
3004  95200.0   1960.0  1332.8  17150.0  63000.0        8190.0  588.0   672.0
3005  11200.0    322.0   156.8   2800.0   8400.0         910.0   56.0    98.0
3006  42000.0   1078.0   588.0   8400.0  24500.0        2730.0  189.0   322.0
3007  95200.0   2128.0  1332.8  17850.0  60200.0        7280.0  574.0   686.0
3008  68600.0   1246.0   960.4  11900.0  49000.0        6370.0  476.0   462.0

n     Riboflavin  Thiamin  Vitamin A  Vitamin B-12  Vitamin B-6  Vitamin C  \
0           46.2     46.2    30800.0         102.9         55.3     2835.0
1            9.1      8.4     6300.0          16.8          9.1      630.0
2           23.8     23.1    15400.0          46.2         25.2     1365.0
3           30.8     30.1    20300.0          60.9         32.9     1750.0
4            0.0      0.0        0.0           0.0          0.0        0.0
…             …       …         …           …            …          …
3004        53.2     51.8    35000.0         102.9         58.8     2800.0
```

```
3005          7.7      7.7    4900.0          16.8          10.5      525.0
3006         25.9     24.5   17500.0          50.4          28.7     1680.0
3007         53.2     51.8   35700.0         107.1          60.2     3150.0
3008         35.7     35.7   23100.0          71.4          38.5     1715.0

n      Zinc
0     378.0
1      77.0
2     175.0
3     245.0
4       0.0
…       …
3004  420.0
3005   56.0
3006  210.0
3007  441.0
3008  280.0

[3009 rows x 15 columns]
```

[24]:
```
proportions = []
for nutrient in consumed_nutrients.columns:
    proportion = required_nutrients_household[nutrient]/
 ↪consumed_nutrients[nutrient]
    proportions.append(proportion)
```

[25]:
```
nutritional_adequancy = pd.DataFrame(proportions).transpose()
nutritional_adequancy.replace(np.inf, 0,inplace=True)
nutritional_adequancy.fillna(0)
nutritional_adequancy = nutritional_adequancy.drop(["Vitamin B-12"], axis = 1)
nutritional_adequancy
```

[25]:
```
         Energy   Protein     Fiber    Folate   Calcium  Carbohydrate  \
0      0.561165  0.856388  0.816054  0.525039  0.820085      0.325675
1      1.020889  0.926471  2.256279  1.535649  1.450944      0.289979
2      0.938697  0.478827  0.783394  2.178218  5.063291      0.431135
3      0.543079  0.852483  0.958435  0.503497  0.755564      0.321153
4      0.000000  0.000000  0.000000  0.000000  0.000000      0.000000
…           …        …        …        …        …            …
3004   0.588667  0.503242  0.817407  0.731323  0.875460      0.365413
3005   2.689281  1.173537  1.072470  0.382667  0.557397      1.065951
3006   0.734575  0.893542  0.481710  0.730484  5.012642      0.253916
3007  12.994458  3.495056  8.206897  0.972911  1.423134      5.161803
3008   0.255863  0.172786  0.303004  0.206890  0.359438      0.160890

          Iron    Niacin  Riboflavin   Thiamin  Vitamin A  Vitamin B-6  \
0     0.642492  0.914105    0.735995  0.563252   0.540391     0.355728
```

```
1     0.629371  0.935004    1.409881  0.590763   0.000000      4.519868
2     0.505114  0.634794    1.058135  0.712189   4.092262      0.658858
3     0.501827  0.877256    0.739354  0.615819   0.415303      0.247935
4     0.000000  0.000000    0.000000  0.000000   0.000000      0.000000
…        …         …         …         …          …          …
3004  0.788975  0.783756    0.639910  0.517727   0.781159      0.344020
3005  0.310488  0.925650    0.612868  1.165507   0.360969      0.457469
3006  0.488263  0.607303    0.835356  0.473288   1.195938      0.347893
3007  1.201289  3.332109    1.620193  4.807068   1.074046      1.235632
3008  0.139255  0.242569    0.213902  0.257914   0.261407      0.183016

      Vitamin C      Zinc
0      0.108545  0.908062
1      0.000000  1.848986
2      1.014779  0.568053
3      0.120349  0.831636
4      0.000000  0.000000
…         …         …
3004   0.107851  0.630329
3005   0.124801  0.685091
3006   0.336645  1.031327
3007   0.315183  2.207428
3008   0.060402  0.207634

[3009 rows x 14 columns]
```

[26]:
```
import matplotlib.pyplot as plt
plt.figure(figsize=(14,6))
plt.bar(nutritional_adequancy.mean().index,nutritional_adequancy.mean() )
plt.axhline(y = 1, color = 'r', linestyle = '-')
plt.title("Nutritional adequacy of hosuehold diet (Uganda 2019 - 2020)")
plt.xticks(fontsize= 9)
plt.ylabel("Ratio of recomended nutrients over actual consumption")
```

```
<IPython.core.display.Javascript object>
```

```
<IPython.core.display.HTML object>
```

[26]:
```
Text(0, 0.5, 'Ratio of recomended nutrients over actual consumption')
```

[ ]:

[ ]: