Cleaning Data Once Again

November 12, 2021

[2]:
```
import pandas as pd
import dataframe_image as dfi
```

0.0.1 Preparing all data

[3]:
```
#Country List
country_list = ['Germany', 'United Kingdom', 'France', 'Italy', 'Russia',␣
 ↪'Spain', 'Netherlands', 'Switzerland', 'Turkey', 'Poland']
```

Legend: VO: Vegetable Oil Consumption

FC: Macronutrient Consumption

CAD: Coronary Artery Disease Mortality

HCD: HealthCareDataIndex

[4]:
```
vo = pd.read_csv('VegetableOilConsumptionPerCapita2007_2011.csv')
mnc = pd.read_csv('top10GDPEurope.csv')
cad = pd.read_csv('HeartDiseaseMortality2007_2011.csv')
hcd = pd.read_csv('formattedHealthCareData.csv')
```

Visualizing Data

[96]:
```
vo.head(3)
```

[96]:
```
  Countries Units        2007        2008        2009        2010        2011
0    France  kcal  507.945205  517.808219  530.136986  520.273973  517.808219
1   Germany  kcal  451.232877  461.095890  446.301370  433.972603  429.041096
2     Italy  kcal  673.150685  675.616438  658.356164  680.547945  690.410959
```

[97]:
```
mnc.head(3)
```

[97]:
```
   Unnamed: 0  Entity Code  Year  Animal protein  Plant protein      Fat  \
0           0  France  FRA  2007          285.96         156.44  1443.69
1           1  France  FRA  2008          285.12         165.20  1505.97
2           2  France  FRA  2009          285.36         159.40  1485.99

   Carbohydrates  Protein
0        1579.91   442.40
```

```
1        1594.71   450.32
2        1599.25   444.76
```

```
[7]: cad.head(3)
```

```
[7]:    Unnamed: 0       Entity Code  Year       Deaths
     0           0  Afghanistan  AFG  2007  707.188774
     1           1  Afghanistan  AFG  2008  693.448663
     2           2  Afghanistan  AFG  2009  677.845507
```

```
[8]: hcd.head(3)
```

```
[8]:    Unnamed: 0      Country  Health Care Index  Infrastructure  Professionals  \
     0           0  South Korea              78.72           87.16          14.23
     1           1       Taiwan              77.70           79.05          13.06
     2           2      Denmark              74.11           90.75          30.01

         Cost  Medicine Availability  Government Readiness
     0  83.59                  82.30                 87.89
     1  78.39                  78.99                 65.09
     2  82.59                  92.06                 96.30
```

What I saw? Need to reshape Vegetable Oil Table

Need to filter data from the country_list

What to do? Filter and the describe data:

Since I chose 5 years and 10 counties there should be 50 items in count when I describe the data

CAD filtering

```
[14]: cad[cad['Entity'].isin(country_list)].describe()
```

```
[14]:         Unnamed: 0         Year      Deaths
      count    50.00000    50.000000   50.000000
      mean    775.00000  2009.000000  196.486640
      std     257.22816     1.428571  135.247507
      min     370.00000  2007.000000  101.182683
      25%     512.25000  2008.000000  125.727912
      50%     829.50000  2009.000000  138.600611
      75%     996.75000  2010.000000  186.434305
      max    1084.00000  2011.000000  608.128850
```

```
[15]: #Saving info in variable
      cad = cad[cad['Entity'].isin(country_list)]
```

Health Care Index Filter

There should be a count of 10 as I am searching the index for 10 countries

[18]:
```
hcd[hcd['Country'].isin(country_list)].describe()
```

[18]:
```
       Unnamed: 0  Health Care Index  Infrastructure  Professionals  \
count   10.000000          10.000000       10.000000      10.000000
mean    27.200000          51.365000       80.164000      17.881000
std     22.493456          11.462612        9.585089       6.847948
min      6.000000          35.960000       64.760000      13.180000
25%      9.250000          40.372500       77.792500      13.570000
50%     16.500000          52.275000       79.010000      14.545000
75%     46.500000          61.337500       85.555000      20.472500
max     64.000000          65.380000       97.400000      34.250000

            Cost  Medicine Availability  Government Readiness
count  10.000000              10.000000              10.00000
mean   68.568000              72.391000              88.18400
std    10.415315              17.698151               5.76143
min    51.700000              47.830000              78.63000
25%    65.575000              58.622500              85.19250
50%    69.260000              68.485000              88.36500
75%    74.662500              88.642500              92.72250
max    87.030000              98.430000              96.80000
```

[19]:
```
#Saving Info
hcd = hcd[hcd['Country'].isin(country_list)]
```

VO filtering

Only 10 columns should appear as data is formated differently

[22]:
```
vo[vo['Countries'].isin(country_list)].describe()
```

[22]:
```
       Unnamed: 0       2007       2008       2009       2010
count   10.000000  10.000000  10.000000  10.000000  10.000000
mean   109.100000  19.100000  19.560000  19.740000  19.950000
std     37.492073   5.564371   5.468333   5.800421   6.050574
min     50.000000  11.200000  11.900000  12.400000  11.800000
25%     80.750000  16.050000  16.725000  15.325000  16.325000
50%    120.500000  18.350000  19.200000  18.950000  19.250000
75%    136.500000  22.925000  22.050000  22.925000  22.525000
max    151.000000  27.300000  27.900000  30.000000  31.000000
```

[23]:
```
vo = vo[vo['Countries'].isin(country_list)]
```

Fixing the VO Table

[29]:
```
vo.reset_index(inplace = True)
```

```
   level_0  index  Unnamed: 0      Countries Units  2007  2008  2009  2010  \
0        0     49          50         France    kg  20.6  21.0  21.5  21.1
1        1     53          54        Germany    kg  18.3  18.7  18.1  17.6
```

```
2        2     71           72            Italy    kg  27.3  27.4  26.7  27.6
3        3    106          107      Netherlands    kg  15.8  16.2  14.4  15.9
4        4    118          119           Poland    kg  11.2  11.9  13.0  11.8
5        5    121          122           Russia    kg  12.1  12.1  12.4  13.0
6        6    131          132            Spain    kg  26.8  27.9  30.0  31.0
7        7    137          138      Switzerland    kg  18.4  19.7  19.6  19.8
8        8    145          146           Turkey    kg  23.7  22.4  23.4  23.0
9        9    150          151   United Kingdom    kg  16.8  18.3  18.3  18.7

   2011
0    21
1  17.4
2    28
3  14.3
4  13.4
5  13.1
6  33.3
7  20.1
8  24.9
9  17.5
```

[31]:

```
vo.drop(['Unnamed: 0'], axis=1, inplace=True)
print(vo)
```

```
        Countries Units  2007  2008  2009  2010  2011
0           France    kg  20.6  21.0  21.5  21.1    21
1          Germany    kg  18.3  18.7  18.1  17.6  17.4
2            Italy    kg  27.3  27.4  26.7  27.6    28
3      Netherlands    kg  15.8  16.2  14.4  15.9  14.3
4           Poland    kg  11.2  11.9  13.0  11.8  13.4
5           Russia    kg  12.1  12.1  12.4  13.0  13.1
6            Spain    kg  26.8  27.9  30.0  31.0  33.3
7      Switzerland    kg  18.4  19.7  19.6  19.8  20.1
8           Turkey    kg  23.7  22.4  23.4  23.0  24.9
9   United Kingdom    kg  16.8  18.3  18.3  18.7  17.5

C:\Users\mauri\anaconda3\lib\site-packages\pandas\core\frame.py:4308:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return super().drop(
```

Converting the Units from kg/year to kcal/day

[41]:

```
vo.iloc[:, 1] = 'kcal'
```

```
C:\Users\mauri\anaconda3\lib\site-packages\pandas\core\indexing.py:1720:
SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

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

[60]:

```
vo.iloc[:, 2 :6] *= 1000
```

```
C:\Users\mauri\anaconda3\lib\site-packages\pandas\core\indexing.py:1754:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self._setitem_single_column(loc, val, pi)
```

[62]:

```
vo.iloc[:, 2 :6] /= 365
```

```
C:\Users\mauri\anaconda3\lib\site-packages\pandas\core\indexing.py:1754:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self._setitem_single_column(loc, val, pi)
```

[64]:

```
vo.iloc[:, 2 :6] *= 9
```

```
C:\Users\mauri\anaconda3\lib\site-packages\pandas\core\indexing.py:1754:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  self._setitem_single_column(loc, val, pi)
```

[65]:

```
vo
```

[65]:

```
     Countries Units        2007        2008        2009        2010  2011
0       France  kcal  507.945205  517.808219  530.136986  520.273973    21
1      Germany  kcal  451.232877  461.095890  446.301370  433.972603  17.4
2        Italy  kcal  673.150685  675.616438  658.356164  680.547945    28
3  Netherlands  kcal  389.589041  399.452055  355.068493  392.054795  14.3
4       Poland  kcal  276.164384  293.424658  320.547945  290.958904  13.4
5       Russia  kcal  298.356164  298.356164  305.753425  320.547945  13.1
```

```
6           Spain  kcal  660.821918  687.945205  739.726027  764.383562  33.3
7     Switzerland  kcal  453.698630  485.753425  483.287671  488.219178  20.1
8          Turkey  kcal  584.383562  552.328767  576.986301  567.123288  24.9
9  United Kingdom  kcal  414.246575  451.232877  451.232877  461.095890  17.5
```

Had an issue with column '2011' as it was a string instead of a float

[70]:

```
vo['2011'] = vo['2011'].astype('float64')
```

```
<ipython-input-70-75375df10274>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  vo['2011'] = vo['2011'].astype('float64')
```

[74]:

```
vo['2011'] *= 1000
```

```
<ipython-input-74-21c143fb44e6>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  vo['2011'] *= 1000
```

[76]:

```
vo['2011'] /= 365
```

```
<ipython-input-76-2cb5011ed6fd>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  vo['2011'] /= 365
```

[79]:

```
vo['2011'] *= 9
```

```
<ipython-input-79-34c91d9dd1a0>:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  vo['2011'] *= 9
```

Succesfully parsed all the column to kcal/days

[80]:

```
vo
```

```
[80]:         Countries Units        2007        2008        2009        2010  \
      0          France  kcal  507.945205  517.808219  530.136986  520.273973
      1         Germany  kcal  451.232877  461.095890  446.301370  433.972603
      2           Italy  kcal  673.150685  675.616438  658.356164  680.547945
      3     Netherlands  kcal  389.589041  399.452055  355.068493  392.054795
      4          Poland  kcal  276.164384  293.424658  320.547945  290.958904
      5          Russia  kcal  298.356164  298.356164  305.753425  320.547945
      6           Spain  kcal  660.821918  687.945205  739.726027  764.383562
      7     Switzerland  kcal  453.698630  485.753425  483.287671  488.219178
      8          Turkey  kcal  584.383562  552.328767  576.986301  567.123288
      9  United Kingdom  kcal  414.246575  451.232877  451.232877  461.095890

               2011
      0  517.808219
      1  429.041096
      2  690.410959
      3  352.602740
      4  330.410959
      5  323.013699
      6  821.095890
      7  495.616438
      8  613.972603
      9  431.506849
```

Reshaping the VO Table The melt function helps me reshape data

```
[136]: vou = pd.melt(vo, id_vars=["Countries"])
```

```
[137]: vou.drop([0,1,2,3,4,5,6,7,8,9], axis=0, inplace=True)
```

```
[138]: vou.reset_index(inplace=True)
```

```
[139]: vou.head()
```

```
[139]:    index    Countries variable       value
       0     10       France     2007  507.945205
       1     11      Germany     2007  451.232877
       2     12        Italy     2007  673.150685
       3     13  Netherlands     2007  389.589041
       4     14       Poland     2007  276.164384
```

```
[140]: vou.drop(['index'], axis=1, inplace=True)
```

```
[141]: vou.rename(columns={'variable':'Year', 'value':'Vegetable Oil'})
```

```
[141]:          Countries  Year Vegetable Oil
       0            France  2007    507.945205
       1           Germany  2007    451.232877
```

```
2            Italy  2007    673.150685
3      Netherlands  2007    389.589041
4           Poland  2007    276.164384
5           Russia  2007    298.356164
6            Spain  2007    660.821918
7      Switzerland  2007     453.69863
8           Turkey  2007    584.383562
9   United Kingdom  2007    414.246575
10          France  2008    517.808219
11         Germany  2008     461.09589
12           Italy  2008    675.616438
13     Netherlands  2008    399.452055
14          Poland  2008    293.424658
15          Russia  2008    298.356164
16           Spain  2008    687.945205
17     Switzerland  2008    485.753425
18          Turkey  2008    552.328767
19  United Kingdom  2008    451.232877
20          France  2009    530.136986
21         Germany  2009     446.30137
22           Italy  2009    658.356164
23     Netherlands  2009    355.068493
24          Poland  2009    320.547945
25          Russia  2009    305.753425
26           Spain  2009    739.726027
27     Switzerland  2009    483.287671
28          Turkey  2009    576.986301
29  United Kingdom  2009    451.232877
30          France  2010    520.273973
31         Germany  2010    433.972603
32           Italy  2010    680.547945
33     Netherlands  2010    392.054795
34          Poland  2010    290.958904
35          Russia  2010    320.547945
36           Spain  2010    764.383562
37     Switzerland  2010    488.219178
38          Turkey  2010    567.123288
39  United Kingdom  2010     461.09589
40          France  2011    517.808219
41         Germany  2011    429.041096
42           Italy  2011    690.410959
43     Netherlands  2011     352.60274
44          Poland  2011    330.410959
45          Russia  2011    323.013699
46           Spain  2011     821.09589
47     Switzerland  2011    495.616438
48          Turkey  2011    613.972603
```

```
49  United Kingdom  2011    431.506849
```

Analyzed all data

We just need to drop all the junk columns to make a new table and rename set the countries column to 'Entity'

VOU = Countries

MNC = Entity (drop Code)

CAD = Entity (drop Code)

HCD = Country (drop Other than Health Care)

Vegetable Oil Done

[142]:

```
vou.rename(columns={'Countries':'Entity', 'variable':'Year', 'value':'Vegetable␣
 ↪Oil'},  inplace=True)
```

[143]:

```
vou.head()
```

[143]:

```
        Entity  Year Vegetable Oil
0       France  2007    507.945205
1      Germany  2007    451.232877
2        Italy  2007    673.150685
3  Netherlands  2007    389.589041
4       Poland  2007    276.164384
```

[144]:

```
#Need to convert to the same datatypes
vou.dtypes
```

[144]:

```
Entity           object
Year             object
Vegetable Oil    object
dtype: object
```

[145]:

```
vou
```

[145]:

```
            Entity  Year Vegetable Oil
0           France  2007    507.945205
1          Germany  2007    451.232877
2            Italy  2007    673.150685
3      Netherlands  2007    389.589041
4           Poland  2007    276.164384
5           Russia  2007    298.356164
6            Spain  2007    660.821918
7      Switzerland  2007     453.69863
8           Turkey  2007    584.383562
9   United Kingdom  2007    414.246575
10          France  2008    517.808219
```

```
11         Germany  2008   461.09589
12           Italy  2008  675.616438
13     Netherlands  2008  399.452055
14          Poland  2008  293.424658
15          Russia  2008  298.356164
16           Spain  2008  687.945205
17     Switzerland  2008  485.753425
18          Turkey  2008  552.328767
19  United Kingdom  2008  451.232877
20          France  2009  530.136986
21         Germany  2009   446.30137
22           Italy  2009  658.356164
23     Netherlands  2009  355.068493
24          Poland  2009  320.547945
25          Russia  2009  305.753425
26           Spain  2009  739.726027
27     Switzerland  2009  483.287671
28          Turkey  2009  576.986301
29  United Kingdom  2009  451.232877
30          France  2010  520.273973
31         Germany  2010  433.972603
32           Italy  2010  680.547945
33     Netherlands  2010  392.054795
34          Poland  2010  290.958904
35          Russia  2010  320.547945
36           Spain  2010  764.383562
37     Switzerland  2010  488.219178
38          Turkey  2010  567.123288
39  United Kingdom  2010   461.09589
40          France  2011  517.808219
41         Germany  2011  429.041096
42           Italy  2011  690.410959
43     Netherlands  2011   352.60274
44          Poland  2011  330.410959
45          Russia  2011  323.013699
46           Spain  2011   821.09589
47     Switzerland  2011  495.616438
48          Turkey  2011  613.972603
49  United Kingdom  2011  431.506849
```

[146]:

```
vou['Vegetable Oil'] = vou['Vegetable Oil'].astype('float64')
```

[148]:

```
vou['Year'] = vou['Year'].astype('int64')
```

[149]:

```
vou.dtypes
```

```
[149]: Entity             object
       Year                int64
       Vegetable Oil     float64
       dtype: object
```

Macronutrient Composition Done

```
[108]: mnc.drop(['Unnamed: 0', 'Code'], axis=1, inplace=True)
```

```
[109]: mnc.head()
```

```
[109]:    Entity  Year  Animal protein  Plant protein      Fat  Carbohydrates  \
       0  France  2007          285.96         156.44  1443.69        1579.91
       1  France  2008          285.12         165.20  1505.97        1594.71
       2  France  2009          285.36         159.40  1485.99        1599.25
       3  France  2010          283.84         168.80  1471.05        1612.31
       4  France  2011          281.64         165.64  1452.42        1614.30

          Protein
       0   442.40
       1   450.32
       2   444.76
       3   452.64
       4   447.28
```

```
[122]: mnc.head().dtypes
```

```
[122]: Entity              object
       Year                 int64
       Animal protein     float64
       Plant protein      float64
       Fat                float64
       Carbohydrates      float64
       Protein            float64
       dtype: object
```

Coronary Artery Disease Done

```
[112]: cad.drop(['Unnamed: 0', 'Code'], axis=1, inplace=True)
```

```
[113]: cad.head()
```

```
[113]:      Entity  Year      Deaths
       370  France  2007  112.366845
       371  France  2008  110.082662
       372  France  2009  107.888487
       373  France  2010  104.153629
       374  France  2011  101.182683
```

Heath Care Index Done

```
[116]: hcd.drop(['Unnamed: 0', 'Infrastructure', 'Professionals', 'Cost', 'Medicine␣
        ↪Availability', 'Government Readiness'], axis=1, inplace=True)
```

```
C:\Users\mauri\anaconda3\lib\site-packages\pandas\core\frame.py:4308:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return super().drop(
```

```
[119]: hcd.rename(columns={'Country':'Entity'}, inplace=True)
```

```
C:\Users\mauri\anaconda3\lib\site-packages\pandas\core\frame.py:4441:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  return super().rename(
```

```
[120]: hcd.head()
```

```
[120]:             Entity  Health Care Index
        6           France              65.38
        7            Spain              64.66
        9   United Kingdom              61.73
        10     Netherlands              60.16
        16         Germany              52.30
```

0.0.2 Time to merge into a table

```
[150]: mergeTable = pd.merge(mnc,vou)
```

```
[152]: mergeTable = pd.merge(mergeTable, hcd)
```

```
[154]: mergeTable = pd.merge(mergeTable, cad)
```

```
[156]: mergeTable.rename(columns={'Fat':'Total Fat', 'Carbohydrates' : 'Total␣
        ↪Carbohydrates', 'Protein' : 'Total Protein'}, inplace=True)
```

```
[157]: mergeTable
```

```
[157]:             Entity  Year  Animal protein  Plant protein  Total Fat  \
        0           France  2007          285.96         156.44    1443.69
        1           France  2008          285.12         165.20    1505.97
        2           France  2009          285.36         159.40    1485.99
```

```
3           France  2010          283.84         168.80    1471.05
4           France  2011          281.64         165.64    1452.42
5          Germany  2007          248.00         161.96    1314.00
6          Germany  2008          245.52         159.60    1296.99
7          Germany  2009          249.36         160.88    1285.29
8          Germany  2010          247.36         160.20    1296.90
9          Germany  2011          248.92         163.84    1310.85
10           Italy  2007          239.52         203.40    1407.60
11           Italy  2008          240.32         202.04    1403.01
12           Italy  2009          239.72         208.32    1374.57
13           Italy  2010          245.04         201.56    1408.50
14           Italy  2011          240.24         199.28    1417.86
15     Netherlands  2007          282.80         141.28    1184.67
16     Netherlands  2008          294.64         137.36    1194.30
17     Netherlands  2009          304.52         138.24    1142.82
18     Netherlands  2010          288.08         142.96    1126.08
19     Netherlands  2011          284.52         143.28    1102.23
20          Poland  2007          201.92         196.44    1006.83
21          Poland  2008          197.64         194.20    1009.62
22          Poland  2009          205.44         197.20    1048.68
23          Poland  2010          209.16         195.88    1054.62
24          Poland  2011          211.72         197.52    1063.08
25          Russia  2007          205.84         188.24     845.73
26          Russia  2008          210.52         187.72     857.88
27          Russia  2009          212.64         185.84     862.47
28          Russia  2010          216.80         187.04     898.20
29          Russia  2011          216.88         188.16     909.27
30           Spain  2007          289.00         155.96    1325.52
31           Spain  2008          272.36         159.96    1297.89
32           Spain  2009          271.40         160.68    1298.52
33           Spain  2010          269.64         158.20    1296.09
34           Spain  2011          265.28         158.24    1299.06
35     Switzerland  2007          239.36         136.72    1400.94
36     Switzerland  2008          239.52         134.16    1427.85
37     Switzerland  2009          239.76         138.80    1410.30
38     Switzerland  2010          243.08         136.12    1412.55
39     Switzerland  2011          240.72         141.72    1426.50
40          Turkey  2007          115.84         294.84    1021.59
41          Turkey  2008          110.64         291.48     992.97
42          Turkey  2009          112.48         290.40    1012.50
43          Turkey  2010          122.20         291.76    1023.84
44          Turkey  2011          129.56         288.28    1085.58
45  United Kingdom  2007          238.20         179.60    1267.56
46  United Kingdom  2008          230.76         180.76    1261.26
47  United Kingdom  2009          230.36         178.00    1257.66
48  United Kingdom  2010          231.72         176.52    1280.79
49  United Kingdom  2011          234.28         177.44    1243.26
```

```
    Total Carbohydrates  Total Protein  Vegetable Oil  Health Care Index  \
0               1579.91         442.40     507.945205              65.38
1               1594.71         450.32     517.808219              65.38
2               1599.25         444.76     530.136986              65.38
3               1612.31         452.64     520.273973              65.38
4               1614.30         447.28     517.808219              65.38
5               1803.04         409.96     451.232877              52.30
6               1814.89         405.12     461.095890              52.30
7               1819.47         410.24     446.301370              52.30
8               1799.54         407.56     433.972603              52.30
9               1814.39         412.76     429.041096              52.30
10              1767.48         442.92     673.150685              44.43
11              1749.63         442.36     675.616438              44.43
12              1776.39         448.04     658.356164              44.43
13              1727.90         446.60     680.547945              44.43
14              1717.62         439.52     690.410959              44.43
15              1628.25         424.08     389.589041              60.16
16              1601.70         432.00     399.452055              60.16
17              1627.42         442.76     355.068493              60.16
18              1650.88         431.04     392.054795              60.16
19              1681.97         427.80     352.602740              60.16
20              1958.81         398.36     276.164384              39.02
21              1944.54         391.84     293.424658              39.02
22              1966.68         402.64     320.547945              39.02
23              1953.34         405.04     290.958904              39.02
24              1970.68         409.24     330.410959              39.02
25              2069.19         394.08     298.356164              37.76
26              2055.88         398.24     298.356164              37.76
27              2012.05         398.48     305.753425              37.76
28              1999.96         403.84     320.547945              37.76
29              2016.69         405.04     323.013699              37.76
30              1452.52         444.96     660.821918              64.66
31              1477.79         432.32     687.945205              64.66
32              1474.40         432.08     739.726027              64.66
33              1459.07         427.84     764.383562              64.66
34              1469.42         423.52     821.095890              64.66
35              1650.98         376.08     453.698630              52.25
36              1643.47         373.68     485.753425              52.25
37              1676.14         378.56     483.287671              52.25
38              1656.25         379.20     488.219178              52.25
39              1682.06         382.44     495.616438              52.25
40              2197.73         410.68     584.383562              35.96
41              2199.91         402.12     552.328767              35.96
42              2205.62         402.88     576.986301              35.96
43              2212.20         413.96     567.123288              35.96
44              2170.58         417.84     613.972603              35.96
```

```
45              1731.64         417.80     414.246575                61.73
46              1749.22         411.52     451.232877                61.73
47              1745.98         408.36     451.232877                61.73
48              1714.97         408.24     461.095890                61.73
49              1762.02         411.72     431.506849                61.73

        Deaths
0   112.366845
1   110.082662
2   107.888487
3   104.153629
4   101.182683
5   176.248974
6   173.195975
7   169.853684
8   164.814325
9   160.844555
10  141.709672
11  138.911587
12  136.287661
13  130.566324
14  129.132829
15  138.289634
16  133.831519
17  127.041622
18  122.263552
19  118.888886
20  302.017225
21  293.687098
22  284.982958
23  268.103396
24  258.747979
25  608.128850
26  601.338551
27  566.176746
28  559.296306
29  513.188675
30  131.780255
31  127.098486
32  120.059699
33  114.388797
34  111.927880
35  132.115958
36  127.292399
37  125.290009
38  120.728443
39  114.813790
```

```
40   187.245539
41   183.640411
42   188.180461
43   186.962429
44   184.849933
45   156.298583
46   151.145975
47   142.040751
48   135.989718
49   129.259572
```

[158]:

```
mergeTable.to_csv('mergedTable.csv')
```

[ ]: