

Achraf Safsafi

DSC 540

Project Milestone3

NOTE: I changed my source website from [https://data.bls.gov to https://www.ssa.gov/](https://data.bls.gov/tohttps://www.ssa.gov/)

```
In [31]: import pandas as pd
import numpy as np
from urllib.request import urlopen
from BeautifulSoup import BeautifulSoup

In [2]: # data for ALABAMA
page = urlopen('https://www.ssa.gov/policy/docs/statcomps/ssi_sc/2018/al.html')
table = BeautifulSoup(page.read())

res = []
row = []

for tr in table.find_all('tr'):
    for th in tr.find_all('th',attrs = ('class':'stub0')):
        row.append(th.text)
    for td in tr.find_all('td'):
        res.append(td.text)
    res.append(row)
    row = []

df_al = pd.DataFrame(data=res)
df_al.insert(0,'State','ALABAMA')
df_al.head()

Out[2]:
State      0      1      2      3      4      5      6      7      8      9
0  ALABAMA  Auburn  01001  1439    60  1379    200  1027  212    909    826
1  ALABAMA  Auburn  02003  305    30  (X)    (X)    (X)    (X)    (X)    (X)    16
2  ALABAMA  Barbour  01005  1395    100  1295    100  888    893  324    557    780
3  ALABAMA  Bibb  01007  896    28    868    74    665    157    350    460
4  ALABAMA  Blount  01009  1227    57  1170  115    887    225    484    654

In [33]: # data for CALIFORNIA
page = urlopen('https://www.ssa.gov/policy/docs/statcomps/ssi_sc/2018/ca.html')
table = BeautifulSoup(page.read())

res = []
row = []

for tr in table.find_all('tr'):
    for th in tr.find_all('th',attrs = ('class':'stub0')):
        row.append(th.text)
    for td in tr.find_all('td'):
        res.append(td.text)
    res.append(row)
    row = []

df_ca = pd.DataFrame(data=res)
df_ca.insert(0,'State','CALIFORNIA')
df_ca.head()

Out[3]:
State      0      1      2      3      4      5      6      7      8      9
0  CALIFORNIA  Alameda  00001  46096  16256  32859  2369    21546  24640  18710  31687
1  CALIFORNIA  Alameda  00003  30    (X)    (X)    (X)    (X)    (X)    (X)    18
2  CALIFORNIA  Alameda  00004  795    95    960    42    900    163    325    439
3  CALIFORNIA  Butte  04007  10748    875  9873    718    7476    2564    4507    6305
4  CALIFORNIA  Calaveras  05009  1000    78    922    80    693    227    444    625

In [41]: # data for GEORGIA
page = urlopen('https://www.ssa.gov/policy/docs/statcomps/ssi_sc/2018/ga.html')
table = BeautifulSoup(page.read())

res = []
row = []

for tr in table.find_all('tr'):
    for th in tr.find_all('th',attrs = ('class':'stub0')):
        row.append(th.text)
    for td in tr.find_all('td'):
        res.append(td.text)
    res.append(row)
    row = []

df_ga = pd.DataFrame(data=res)
df_ga.insert(0,'State','GEORGIA')
df_ga.head()

Out[4]:
State      0      1      2      3      4      5      6      7      8      9
0  GEORGIA  Appling  13001  642    50    592    85  415  142  245    398
1  GEORGIA  Augusta  13003  373    32    341    47  244    70  113    256
2  GEORGIA  Baldwin  13005  378    10    368    60  244    74  138    208
3  GEORGIA  Baker  13007  139    14    125    10    94    65    62    67
4  GEORGIA  Baldwin  13009  1512  106  1406  246  999  267  542    869

In [51]: # data for ILLINOIS
page = urlopen('https://www.ssa.gov/policy/docs/statcomps/ssi_sc/2018/il.html')
table = BeautifulSoup(page.read())

res = []
row = []

for tr in table.find_all('tr'):
    for th in tr.find_all('th',attrs = ('class':'stub0')):
        row.append(th.text)
    for td in tr.find_all('td'):
        res.append(td.text)
    res.append(row)
    row = []

df_il = pd.DataFrame(data=res)
df_il.insert(0,'State','ILLINOIS')
df_il.head()

Out[5]:
State      0      1      2      3      4      5      6      7      8      9
0  ILLINOIS  Adams  18001  279    16    363    47    275    44  128    229
1  ILLINOIS  Adams  18003  363    10    403    747  1480  5324  1126  2489  4769
2  ILLINOIS  Bond  17005  327    11    316    31  250    46  130    186
3  ILLINOIS  Boone  17007  577    69  508  104  371  102  160  330
4  ILLINOIS  Brown  17009  72    (X)    (X)    (X)    53    (X)    31    37

In [61]: # data for INDIANA
page = urlopen('https://www.ssa.gov/policy/docs/statcomps/ssi_sc/2018/in.html')
table = BeautifulSoup(page.read())

res = []
row = []

for tr in table.find_all('tr'):
    for th in tr.find_all('th',attrs = ('class':'stub0')):
        row.append(th.text)
    for td in tr.find_all('td'):
        res.append(td.text)
    res.append(row)
    row = []

df_in = pd.DataFrame(data=res)
df_in.insert(0,'State','INDIANA')
df_in.head()

Out[6]:
State      0      1      2      3      4      5      6      7      8      9
0  INDIANA  Adams  18001  279    16    363    47    275    44  128    229
1  INDIANA  Adams  18003  363    10    403    747  1480  5324  1126  2489  4769
2  INDIANA  Bartholomew  18005  1233    50  1173    192    882    149  431    682
3  INDIANA  Benton  18007  171    (X)    (X)    29  119  33  58    95
4  INDIANA  Blufford  18009  254  12  242    36  194  24  92    151

In [71]: # data for KANSAS
page = urlopen('https://www.ssa.gov/policy/docs/statcomps/ssi_sc/2018/ks.html')
table = BeautifulSoup(page.read())

res = []
row = []

for tr in table.find_all('tr'):
    for th in tr.find_all('th',attrs = ('class':'stub0')):
        row.append(th.text)
    for td in tr.find_all('td'):
        res.append(td.text)
    res.append(row)
    row = []

df_ks = pd.DataFrame(data=res)
df_ks.insert(0,'State','KANSAS')
df_ks.head()

Out[7]:
State      0      1      2      3      4      5      6      7      8      9
0  KANSAS  Allen  20001  363  12  351    53  383  47  351  195
1  KANSAS  Anderson  20003  132    (X)    (X)    25  15  15  11  72
2  KANSAS  Anthony  20005  341  13  328  63  239  39  129  204
3  KANSAS  Barber  20007  66    (X)    (X)    (X)    46    (X)    29  33
4  KANSAS  Barton  20009  495  16  479  60  376  59  200  279

In [81]: # data for KENTUCKY
page = urlopen('https://www.ssa.gov/policy/docs/statcomps/ssi_sc/2018/ky.html')
table = BeautifulSoup(page.read())

res = []
row = []

for tr in table.find_all('tr'):
    for th in tr.find_all('th',attrs = ('class':'stub0')):
        row.append(th.text)
    for td in tr.find_all('td'):
        res.append(td.text)
    res.append(row)
    row = []

df_ky = pd.DataFrame(data=res)
df_ky.insert(0,'State','KENTUCKY')
df_ky.head()

Out[8]:
State      0      1      2      3      4      5      6      7      8      9
0  KENTUCKY  Allen  21003  321    68    863  187    826  716  390  521
1  KENTUCKY  Allen  21005  786  34  782    80  565  141  318  396
2  KENTUCKY  Anderson  21009  429  24  405  75  294  70  158  232
3  KENTUCKY  Ballard  21007  272  13  259  31  199  42  100  140
4  KENTUCKY  Barron  21009  1075  111  1564  199  1136  340  682  873

In [91]: # data for LOUISIANA
page = urlopen('https://www.ssa.gov/policy/docs/statcomps/ssi_sc/2018/la.html')
table = BeautifulSoup(page.read())

res = []
row = []

for tr in table.find_all('tr'):
    for th in tr.find_all('th',attrs = ('class':'stub0')):
        row.append(th.text)
    for td in tr.find_all('td'):
        res.append(td.text)
    res.append(row)
    row = []

df_la = pd.DataFrame(data=res)
df_la.insert(0,'State','LOUISIANA')
df_la.head()

Out[9]:
State      0      1      2      3      4      5      6      7      8      9
0  LOUISIANA  Acadia  22001  2362  103  2401  898  1065  398  1418
1  LOUISIANA  Allen  22003  702  53  709    81  500  161  396  387
2  LOUISIANA  Ascension  22005  2094  131  1963  446  1277  361  629  1190
3  LOUISIANA  Assumption  22007  885  64  821  152  357  216  346  476
4  LOUISIANA  Avoyelles  22009  2461  133  2318  428  1506  518  854  1351

In [101]: # data for MICHIGAN
page = urlopen('https://www.ssa.gov/policy/docs/statcomps/ssi_sc/2018/mi.html')
table = BeautifulSoup(page.read())

res = []
row = []

for tr in table.find_all('tr'):
    for th in tr.find_all('th',attrs = ('class':'stub0')):
        row.append(th.text)
    for td in tr.find_all('td'):
        res.append(td.text)
    res.append(row)
    row = []

df_mi = pd.DataFrame(data=res)
df_mi.insert(0,'State','MICHIGAN')
df_mi.head()

Out[10]:
State      0      1      2      3      4      5      6      7      8      9
0  MICHIGAN  Adams  26001  331  18  313  37  269  45  130  358
1  MICHIGAN  Alger  26003  154  13  141  13  99  42  71  92
2  MICHIGAN  Alcona  26005  1638  57  1581  231  1184  232  562  825
3  MICHIGAN  Alpena  26007  1138  39  1099  124  883  131  463  630
4  MICHIGAN  Antrim  26009  481  13  468  64  362  55  173  295

In [111]: # data for MINNESOTA
page = urlopen('https://www.ssa.gov/policy/docs/statcomps/ssi_sc/2018/mn.html')
table = BeautifulSoup(page.read())

res = []
row = []

for tr in table.find_all('tr'):
    for th in tr.find_all('th',attrs = ('class':'stub0')):
        row.append(th.text)
    for td in tr.find_all('td'):
        res.append(td.text)
    res.append(row)
    row = []

df_mn = pd.DataFrame(data=res)
df_mn.insert(0,'State','MINNESOTA')
df_mn.head()

Out[11]:
State      0      1      2      3      4      5      6      7      8      9
0  MINNESOTA  Ancker  27001  518  18  515  271  36  305  45  103  358
1  MINNESOTA  Anoka  27003  4461  168  3903  737  2795  1018  1089  2792
2  MINNESOTA  Becker  27005  662  58  604  108  415  139  255  375
3  MINNESOTA  Beltrami  27007  1222  79  1143  170  832  220  483  698
4  MINNESOTA  Benton  27009  387  18  369  60  284  43  140  208

In [121]: # data for MISSOURI
page = urlopen('https://www.ssa.gov/policy/docs/statcomps/ssi_sc/2018/mo.html')
table = BeautifulSoup(page.read())

res = []
row = []

for tr in table.find_all('tr'):
    for th in tr.find_all('th',attrs = ('class':'stub0')):
        row.append(th.text)
    for td in tr.find_all('td'):
        res.append(td.text)
    res.append(row)
    row = []

df_mo = pd.DataFrame(data=res)
df_mo.insert(0,'State','MISSOURI')
df_mo.head()

Out[12]:
State      0      1      2      3      4      5      6      7      8      9
0  MISSOURI  Ash  29001  629  24  588  72  466  84  263  300
1  MISSOURI  Ash Grove  29003  140  (X)    (X)    23  114  35  65  96
2  MISSOURI  Atchison  29005  79  (X)    (X)    (X)    51  18  32  39
3  MISSOURI  Audrain  29007  573  27  546  78  423  72  225  334
4  MISSOURI  Barry  29009  945  55  890  102  692  151  368  516

In [131]: # data for NEBRASKA
page = urlopen('https://www.ssa.gov/policy/docs/statcomps/ssi_sc/2018/ne.html')
table = BeautifulSoup(page.read())

res = []
row = []

for tr in table.find_all('tr'):
    for th in tr.find_all('th',attrs = ('class':'stub0')):
        row.append(th.text)
    for td in tr.find_all('td'):
        res.append(td.text)
    res.append(row)
    row = []

df_ne = pd.DataFrame(data=res)
df_ne.insert(0,'State','NEBRASKA')
df_ne.head()

Out[13]:
State      0      1      2      3      4      5      6      7      8      9
0  NEBRASKA  Adams  31001  546  43  500  64  354  96  219  261
1  NEBRASKA  Artesia  31003  85  (X)    (X)    (X)    63  (X)    32  45
2  NEBRASKA  Arthur  31005  (X)    (X)    (X)    (X)    (X)    (X)    (X)    (X)
3  NEBRASKA  Banner  31007  (X)    (X)    (X)    (X)    (X)    (X)    (X)    (X)
4  NEBRASKA  Blaine  31009  (X)    (X)    (X)    (X)    (X)    (X)    (X)    (X)

In [141]: # data for NEW YORK
page = urlopen('https://www.ssa.gov/policy/docs/statcomps/ssi_sc/2018/ny.html')
table = BeautifulSoup(page.read())

res = []
row = []

for tr in table.find_all('tr'):
    for th in tr.find_all('th',attrs = ('class':'stub0')):
        row.append(th.text)
    for td in tr.find_all('td'):
        res.append(td.text)
    res.append(row)
    row = []

df_ny = pd.DataFrame(data=res)
df_ny.insert(0,'State','NEW YORK')
df_ny.head()

Out[14]:
State      0      1      2      3      4      5      6      7      8      9
0  NEW YORK  Albany  36001  7174  615  6589  1138  4669  1370  2378  4183
1  NEW YORK  Albany  36003  1343  52  1291  165  975  203  439  773
2  NEW YORK  Broome  36005  104324  14435  89389  17488  51260  35576  29821  63433
3  NEW YORK  Broome  36007  59956  352  6743  1326  4746  1024  1204  4277
4  NEW YORK  Cattaraugus  36009  2405  64  2321  301  1775  329  890  1418

In [151]: # data for NORTH DAKOTA
page = urlopen('https://www.ssa.gov/policy/docs/statcomps/ssi_sc/2018/nd.html')
table = BeautifulSoup(page.read())

res = []
row = []

for tr in table.find_all('tr'):
    for th in tr.find_all('th',attrs = ('class':'stub0')):
        row.append(th.text)
    for td in tr.find_all('td'):
        res.append(td.text)
    res.append(row)
    row = []

df_nd = pd.DataFrame(data=res)
df_nd.insert(0,'State','NORTH DAKOTA')
df_nd.head()

Out[15]:
State      0      1      2      3      4      5      6      7      8      9
0  NORTH DAKOTA  Adams  38003  17  (X)    17  (X)    (X)    (X)    (X)    6
1  NORTH DAKOTA  Barnes  38005  125  (X)    (X)    (X)    92  (X)    40  86
2  NORTH DAKOTA  Benson  38007  173  13  160  24  109  40  66  99
3  NORTH DAKOTA  Billings  38009  (X)    (X)    (X)    (X)    (X)    (X)    (X)    (X)
4  NORTH DAKOTA  Bottineau  38009  46  (X)    (X)    (X)    31  (X)    23  21

In [161]: # data for OHIO
page = urlopen('https://www.ssa.gov/policy/docs/statcomps/ssi_sc/2018/oh.html')
table = BeautifulSoup(page.read())

res = []
row = []

for tr in table.find_all('tr'):
    for th in tr.find_all('th',attrs = ('class':'stub0')):
        row.append(th.text)
    for td in tr.find_all('td'):
        res.append(td.text)
    res.append(row)
    row = []

df_oh = pd.DataFrame(data=res)
df_oh.insert(0,'State','OHIO')
df_oh.head()

Out[16]:
State      0      1      2      3      4      5      6      7      8      9
0  OHIO  Adams  39001  1718  57  1661  156  1271  291  526  1081
1  OHIO  Allen  39003  2977  107  2870  386  2196  405  940  1391
2  OHIO  Ashland  39005  794  21  773  129  385  848  242  494
3  OHIO  Ashtabula  39007  3054  80  2974  372  2394  385  867  1781
4  OHIO  Athens  39009  2433  74  2539  240  1883  390  767  1547

In [171]: # data for OKLAHOMA
page = urlopen('https://www.ssa.gov/policy/docs/statcomps/ssi_sc/2018/ok.html')
table = BeautifulSoup(page.read())

res = []
row = []

for tr in table.find_all('tr'):
    for th in tr.find_all('th',attrs = ('class':'stub0')):
        row.append(th.text)
    for td in tr.find_all('td'):
        res.append(td.text)
    res.append(row)
    row = []

df_ok = pd.DataFrame(data=res)
df_ok.insert(0,'State','OKLAHOMA')
df_ok.head()

Out[17]:
State      0      1      2      3      4      5      6      7      8      9
0  OKLAHOMA  Adair  40001  927  61  866  129  621  177  319  548
1  OKLAHOMA  Alfalfa  40003  73  (X)    (X)    (X)    56  (X)    26  37
2  OKLAHOMA  Atoka  40005  539  49  490  62  363  114  215  299
3  OKLAHOMA  Beaver  40007  43  (X)    (X)    (X)    29  (X)    12  25
4  OKLAHOMA  Beckham  40009  522  39  483  68  351  103  222  287

In [181]: # data for PENNSYLVANIA
page = urlopen('https://www.ssa.gov/policy/docs/statcomps/ssi_sc/2018/pa.html')
table = BeautifulSoup(page.read())

res = []
row = []

for tr in table.find_all('tr'):
    for th in tr.find_all('th',attrs = ('class':'stub0')):
        row.append(th.text)
    for td in tr.find_all('td'):
        res.append(td.text)
    res.append(row)
    row = []

df_pa = pd.DataFrame(data=res)
df_pa.insert(0,'State','PENNSYLVANIA')
df_pa.head()

Out[18]:
State      0      1      2      3      4      5      6      7      8      9
0  PENNSYLVANIA  Adams  42001  1114  89  1045  215  1478  382  676  1156
1  PENNSYLVANIA  Allegheny  42003  32363  1976  3077  5539  20394  6430  9383  19378
2  PENNSYLVANIA  Armstrong  42005  2001  61  1940  238  1455  308  719  1130
3  PENNSYLVANIA  Beaver  42007  4461  184  4277  778  3045  638  1437  2342
4  PENNSYLVANIA  Bedford  42009  1231  78  1153  164  820  247  485  669

In [191]: # data for SOUTH DAKOTA
page = urlopen('https://www.ssa.gov/policy/docs/statcomps/ssi_sc/2018/sd.html')
table = BeautifulSoup(page.read())

res = []
row = []

for tr in table.find_all('tr'):
    for th in tr.find_all('th',attrs = ('class':'stub0')):
        row.append(th.text)
    for td in tr.find_all('td'):
        res.append(td.text)
    res.append(row)
    row = []

df_sd = pd.DataFrame(data=res)
df_sd.insert(0,'State','SOUTH DAKOTA')
df_sd.head()

Out[19]:
State      0      1      2      3      4      5      6      7      8      9
0  SOUTH DAKOTA  Aurora  46003  17  (X)    (X)    (X)    50  (X)    0  7
1  SOUTH DAKOTA  Beadle  46005  406  57  349  56  258  92  145  220
2  SOUTH DAKOTA  Bennett  46007  138  (X)    (X)    21  91  26  43  71
3  SOUTH DAKOTA  Bon Homme  46009  79  17  62  13  38  26  22  37
4  SOUTH DAKOTA  Brookings  46011  278  32  246  33  191  54  92  151

In [201]: # data for TENNESSEE
page = urlopen('https://www.ssa.gov/policy/docs/statcomps/ssi_sc/2018/tn.html')
table = BeautifulSoup(page.read())

res = []
row = []

for tr in table.find_all('tr'):
    for th in tr.find_all('th',attrs = ('class':'stub0')):
        row.append(th.text)
    for td in tr.find_all('td'):
        res.append(td.text)
    res.append(row)
    row = []

df_tn = pd.DataFrame(data=res)
df_tn.insert(0,'State','TENNESSEE')
df_tn.head()

Out[20]:
State      0      1      2      3      4      5      6      7      8      9
0  TENNESSEE  Anderson  47001  2075  77  1598  215  1478  382  676  1156
1  TENNESSEE  Bedford  47003  1083  49  1044  150  799  144  351  600
2  TENNESSEE  Benton  47005  579  31  548  45  428  106  228  309
3  TENNESSEE  Blount  47009  339  33  366  24  275  100  177  202
4  TENNESSEE  Blount  47009  2469  131  2338  242  1741  486  815  1373

In [211]: # data for TEXAS
page = urlopen('https://www.ssa.gov/policy/docs/statcomps/ssi_sc/2018/tx.html')
table = BeautifulSoup(page.read())

res = []
row = []

for tr in table.find_all('tr'):
    for th in tr.find_all('th',attrs = ('class':'stub0')):
        row.append(th.text)
    for td in tr.find_all('td'):
        res.append(td.text)
    res.append(row)
    row = []

df_tx = pd.DataFrame(data=res)
df_tx.insert(0,'State','TEXAS')
df_tx.head()

Out[21]:
State      0      1      2      3      4      5      6      7      8      9
0  TEXAS  Anderson  48001  1328  126  1403  225  1015  289  327  833
1  TEXAS  Andrews  48003  260  39  221  33  145  82  99  134
2  TEXAS  Anglin  48005  3,109  200  2,909  694  1,684  521  1,032  1,735
3  TEXAS  Aransas  48007  506  44  462  54  344  108  164  272
4  TEXAS  Archer  48009  110  (X)    (X)    20  72  18  43  67

In [221]: # data for VIRGINIA
page = urlopen('https://www.ssa.gov/policy/docs/statcomps/ssi_sc/2018/va.html')
table = BeautifulSoup(page.read())

res = []
row = []

for tr in table.find_all('tr'):
    for th in tr.find_all('th',attrs = ('class':'stub0')):
        row.append(th.text)
    for td in tr.find_all('td'):
        res.append(td.text)
    res.append(row)
    row = []

df_va = pd.DataFrame(data=res)
df_va.insert(0,'State','VIRGINIA')
df_va.head()

Out[22]:
State      0      1      2      3      4      5      6      7      8      9
0  VIRGINIA  Accomack  51001  1072  101  971  187  674  211  431  588
1  VIRGINIA  Albemarle  51003  1,009  104  905  142  655  218  351  600
2  VIRGINIA  Alleghany  51005  307  13  294  42  205  60  107  161
3  VIRGINIA  Amelia  51007  306  26  280  41  204  61  99  170
4  VIRGINIA  Amherst  51009  826  46  780  121  573  132  264  464

In [231]: # append all the data frames
all_df = (df_al,df_ga,df_il,df_in,df_ks,
df_ky,df_la,df_mi,df_mo,df_nd,df_ny,df_ok,df_oh,df_pa,df_sd,df_tx,df_va)

df = pd.concat(all_df, ignore_index=True, sort=True)

Out[23]:
State      0      1      2      3      4      5      6      7      8      9
0  ALABAMA  Auburn  01001  1439    60  1379    200  1027  212    909    826
1  ALABAMA  Baldwin  01003  305    30  (X)    (X)    (X)    (X)    (X)    (X)    16
2  ALABAMA  Barbour  01005  1395    100  1295    100  888    893  324    557    780
3  ALABAMA  Bibb  01007  896    28    868    74    665    157    350    460
4  ALABAMA  Blount  01009  1227    57  1170  115    887    225    484    654
... ..
2035  VIRGINIA  Suffolk  51800  2356  132  2,224  336  1,616  404  758  1,366
2036  VIRGINIA  Virginia Beach  51810  5,610  627  4,983  929  3,583  1,098  1,576  3,251
2037  VIRGINIA  Williamsburg  51820  730  32  698  112  518  100  216  433
2038  VIRGINIA  Williamsburg  51830  (X)    (X)    (X)    (X)    (X)    (X)    (X)    (X)
2039  VIRGINIA  Winchester  51840  735  34  701  135  496  104  239  436
2040 rows x 11 columns

In [241]: # remove column headers
df.columns = ['State','County','ANSI Code','Total',
'Aged','Blind_and_disabled','Age_Under18','Age_18-64',
'65_or_older','SSI_recipients_also_receiving_OASDI',
'ds_of_dollars']
df.head()

Out[24]:
State County ANSI Code Total Aged Blind_and_disabled Age_Under18 Age_18-64 65_or_older SSI_recipients_also_recei
ds_of_dollars
0 ALABAMA Auburn 01001 1439 60 1379 200 1027 212 509 826
1 ALABAMA Baldwin 01003 305 200 305 599 2345 561
2 ALABAMA Barbour 01005 1395 100 1295 188 883 324
3 ALABAMA Bibb 01007 896 28 896 74 665 157
4 ALABAMA Blount 01009 1227 57 1170 115 887 225
... ..
1632 VIRGINIA Staunton 51790 825 50 775 101 566 158
1633 VIRGINIA Suffolk 51800 2356 132 2224 336 1616 404
1634 VIRGINIA Virginia Beach 51810 5610 627 4983 929 3583 1098
1635 VIRGINIA Waynesboro 51820 730 32 698 112 518 100
1636 VIRGINIA Winchester 51840 735 34 701 135 496 104
1637 rows x 11 columns

In [251]: # convert integers into int
df['Total'] = pd.to_numeric(df['Total'])
df['Aged'] = pd.to_numeric(df['Aged'])
df['Blind_and_disabled'] = pd.to_numeric(df['Blind_and_disabled'])
df['Age_Under18'] = pd.to_numeric(df['Age_Under18'])
df['Age_18-64'] = pd.to_numeric(df['Age_18-64'])
df['65_or_older'] = pd.to_numeric(df['65_or_older'])
df['SSI_recipients_also_receiving_OASDI'] = pd.to_numeric(df['SSI_recipients_also_receiving_OASDI'])
df['Amount_of_payments(thousands_of_dollars)'] = pd.to_numeric(df['Amount_of_payments(thousands_of_dollars)'])

In [261]: #Check the data type again
df.dtypes

Out[26]:
State County ANSI Code object
Total object
Aged int64
Blind_and_disabled int64
Age_Under18 int64
Age_18-64 int64
65_or_older int64
SSI_recipients_also_receiving_OASDI int64
Amount_of_payments(thousands_of_dollars) int64
dtype: object

In [271]: # remove whitespace from the beginning and end of columns (e.g. strip())
for x in df.columns:
df[x] = df[x].strip()

In [281]: #checking for missing values
df.isna().sum()

Out[28]:
State 0
County 0
ANSI Code 0
Total 0
Aged 0
Blind_and_disabled 0
Age_Under18 0
Age_18-64 0
65_or_older 0
SSI_recipients_also_receiving_OASDI 0
Amount_of_payments(thousands_of_dollars) 0
dtype: int64

In [291]: df.isnull()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3637 entries, 0 to 3636
data columns (total 11 columns):
# 0 1 2 3 4 5 6 7 8 9
... ..
6 State 1637 non-null object
7 County 1637 non-null object
8 ANSI Code 1637 non-null object
9 Total 1637 non-null int64
10 Aged 1637 non-null int64
11 Blind_and_disabled 1637 non-null int64
12 Age_Under18 1637 non-null int64
13 Age_18-64 1637 non-null int64
14 65_or_older 1637 non-null int64
15 SSI_recipients_also_receiving_OASDI 1637 non-null int64
16 Amount_of_payments(thousands_of_dollars) 1637 non-null int64
dtypes: int64(15), object(2)
```