rebecca_notebook

December 3, 2018

In [3]:
```
import pandas as pd
```

In [4]:
```
# Read the Excel workbook
df = pd.read_excel('data/2018-usda-food-environment-atlas-dataset.xls')
```

In [5]:
```
df.head()
```

Out[5]:

	Food Environment Atlas data download
0	Notes about the Food Environment Atlas downloa...
1	This file contains multiple spreadsheets:
2	1. A variable list that includes metadata abo...
3	2. Spreadsheets that contain data for each of...
4	3. County and State-level supplemental data t...

In [6]:
```
# Load Access, Stores, Assistance, Insecurity, Local, Health, Restaurants, Socioeconomic

access = pd.read_excel('data/2018-usda-food-environment-atlas-dataset.xls', 'ACCESS')
stores = pd.read_excel('data/2018-usda-food-environment-atlas-dataset.xls', 'STORES')
assistance = pd.read_excel('data/2018-usda-food-environment-atlas-dataset.xls', 'ASSISTA
insecurity = pd.read_excel('data/2018-usda-food-environment-atlas-dataset.xls', 'INSECUR
local = pd.read_excel('data/2018-usda-food-environment-atlas-dataset.xls', 'LOCAL')
health = pd.read_excel('data/2018-usda-food-environment-atlas-dataset.xls', 'HEALTH')
restaurants = pd.read_excel('data/2018-usda-food-environment-atlas-dataset.xls', 'RESTAU
socioeconomic = pd.read_excel('data/2018-usda-food-environment-atlas-dataset.xls', 'SOCI
prices_taxes = pd.read_excel('data/2018-usda-food-environment-atlas-dataset.xls', 'PRICE
```

In [13]:
```
```

In [19]:
```
restaurants.head()
restaurants_MA = restaurants[restaurants['State'] == 'MA']
restaurant_cols = ['FIPS', 'State', 'County', 'FFRPTH14', 'FSRPTH14']
restaurants_MA = restaurants_MA[restaurant_cols]
restaurants_MA
```

Out[19]:

	FIPS	State	County	FFRPTH14	FSRPTH14
1217	25001	MA	Barnstable	1.005053	1.986841
1218	25003	MA	Berkshire	0.916754	1.670357
1219	25005	MA	Bristol	0.696507	0.844470
1220	25007	MA	Dukes	1.382807	3.457018

```
1221  25009    MA       Essex  0.748936  0.807447
1222  25011    MA    Franklin  0.493918  0.917276
1223  25013    MA     Hampden  0.625853  0.715566
1224  25015    MA   Hampshire  0.615140  1.043874
1225  25017    MA   Middlesex  0.772456  0.829770
1226  25019    MA   Nantucket  1.565954  4.974208
1227  25021    MA     Norfolk  0.680386  0.868178
1228  25023    MA    Plymouth  0.603524  0.808643
1229  25025    MA     Suffolk  0.970995  1.196475
1230  25027    MA   Worcester  0.714220  0.705615
```

In [20]:
```
stores_MA = stores[stores['State'] == 'MA']
stores_cols = ['FIPS', 'State', 'County', 'GROCPTH14', 'SUPERCPTH14', 'CONVSPTH14', 'SP
stores_MA = stores_MA[stores_cols]
stores_MA
```

Out[20]:
```
       FIPS State      County  GROCPTH14  SUPERCPTH14  CONVSPTH14  SPECSPTH14  \
1217  25001    MA  Barnstable   0.348977     0.004653    0.581628    0.251263
1218  25003    MA   Berkshire   0.264149     0.007769    0.536068    0.100998
1219  25005    MA     Bristol   0.187660     0.012631    0.481781    0.075786
1220  25007    MA       Dukes   0.691404     0.000000    0.460936    0.633787
1221  25009    MA       Essex   0.243144     0.003901    0.357565    0.100118
1222  25011    MA    Franklin   0.282239     0.014112    0.493918    0.098784
1223  25013    MA     Hampden   0.209330     0.008544    0.508372    0.061944
1224  25015    MA   Hampshire   0.229901     0.006214    0.316890    0.074562
1225  25017    MA   Middlesex   0.182766     0.005731    0.362348    0.083423
1226  25019    MA   Nantucket   0.368460     0.000000    0.552690    0.552690
1227  25021    MA     Norfolk   0.171902     0.010112    0.365473    0.082340
1228  25023    MA    Plymouth   0.167646     0.005917    0.455602    0.102560
1229  25025    MA     Suffolk   0.282827     0.001303    0.431409    0.099055
1230  25027    MA   Worcester   0.167184     0.009834    0.413043    0.057777

      SNAPSPTH16
1217    0.833038
1218    0.742037
1219    0.881209
1220    0.497700
1221    0.696817
1222    0.736457
1223    1.020705
1224    0.606141
1225    0.522978
1226    0.363372
1227    0.489710
1228    0.618552
1229    0.864011
1230    0.742344
```

In [21]:
```
access_MA = access[access['State'] == 'MA']
```

```
access_cols = ['FIPS', 'State', 'County', 'PCT_LACCESS_POP15', 'PCT_LACCESS_LOWI15', 'P
access_MA = access_MA[access_cols]
access_MA
```

Out[21]:

	FIPS	State	County	PCT_LACCESS_POP15	PCT_LACCESS_LOWI15	\
1217	25001	MA	Barnstable	48.379442	9.717726	
1218	25003	MA	Berkshire	24.321984	5.971745	
1219	25005	MA	Bristol	28.955995	4.502100	
1220	25007	MA	Dukes	7.120600	1.372931	
1221	25009	MA	Essex	29.959103	4.413228	
1222	25011	MA	Franklin	15.492427	5.682865	
1223	25013	MA	Hampden	22.289413	4.932812	
1224	25015	MA	Hampshire	38.655972	9.980293	
1225	25017	MA	Middlesex	24.825819	2.993366	
1226	25019	MA	Nantucket	10.422783	1.150662	
1227	25021	MA	Norfolk	31.694543	3.476869	
1228	25023	MA	Plymouth	40.820126	5.824657	
1229	25025	MA	Suffolk	0.750733	0.211504	
1230	25027	MA	Worcester	28.365209	5.020362	

	PCT_LACCESS_HHNV15	PCT_LACCESS_SNAP15	PCT_LACCESS_CHILD15	\
1217	2.312366	3.192731	8.717733	
1218	2.486676	2.623480	4.601347	
1219	1.451063	2.184273	6.523161	
1220	1.065238	0.137535	1.459410	
1221	1.520608	1.926098	7.005516	
1222	2.958759	2.178276	3.040781	
1223	1.822251	2.422411	4.661253	
1224	3.888760	2.737564	5.955018	
1225	1.087936	1.062284	6.147123	
1226	1.222215	0.301460	2.461146	
1227	1.551223	1.520745	8.180235	
1228	1.814911	2.746281	10.072809	
1229	0.146702	0.126259	0.144534	
1230	2.014543	2.523371	6.735784	

	PCT_LACCESS_SENIORS15	PCT_LACCESS_WHITE15	PCT_LACCESS_BLACK15	\
1217	11.469996	45.383442	0.705073	
1218	4.731523	22.841151	0.482051	
1219	3.940878	27.101606	0.569744	
1220	1.248538	6.207773	0.155550	
1221	4.312460	27.287571	0.513947	
1222	2.103533	14.261431	0.294346	
1223	3.765549	19.934845	0.949074	
1224	4.657704	33.119500	1.242268	
1225	3.336203	21.883684	0.465024	
1226	0.715661	8.843923	1.051898	
1227	4.323727	28.006690	1.306877	

```
1228              5.695550            38.532252             0.735054
1229              0.116408             0.547030             0.098940
1230              3.808896            25.709256             0.680676

      PCT_LACCESS_HISP15  PCT_LACCESS_NHASIAN15  PCT_LACCESS_NHNA15  \
1217            0.910748               0.465696            0.282208
1218            0.561529               0.397395            0.026324
1219            0.624858               0.505494            0.041403
1220            0.205257               0.051552            0.052324
1221            1.729599               0.915813            0.054100
1222            0.627082               0.304809            0.052119
1223            1.648908               0.429622            0.042119
1224            2.113138               2.398717            0.090623
1225            0.838472               1.656166            0.030272
1226            1.177015               0.113373            0.000000
1227            0.908682               1.501928            0.042983
1228            0.618556               0.374305            0.088313
1229            0.099973               0.035544            0.001833
1230            1.389835               0.990059            0.048654

      PCT_LACCESS_NHPI15  PCT_LACCESS_MULTIR15
1217            0.014066              1.528957
1218            0.003733              0.571331
1219            0.009422              0.728326
1220            0.024191              0.629211
1221            0.005655              1.182018
1222            0.002802              0.576919
1223            0.005658              0.928095
1224            0.013593              1.791272
1225            0.005785              0.784888
1226            0.000000              0.413589
1227            0.003747              0.832317
1228            0.009618              1.080585
1229            0.000296              0.067089
1230            0.006306              0.930258
```

In [22]:
```
assistance_MA = assistance[assistance['State'] == 'MA']
assistance_cols = ['FIPS', 'State', 'County', 'PCT_SNAP16', 'PCT_NSLP15', 'PCT_SBP15',
assistance_MA = assistance_MA[assistance_cols]
assistance_MA
```

Out[22]:

	FIPS	State	County	PCT_SNAP16	PCT_NSLP15	PCT_SBP15	PCT_SFSP15	\
1217	25001	MA	Barnstable	11.363419	7.586022	2.489746	0.87744	
1218	25003	MA	Berkshire	11.363419	7.586022	2.489746	0.87744	
1219	25005	MA	Bristol	11.363419	7.586022	2.489746	0.87744	
1220	25007	MA	Dukes	11.363419	7.586022	2.489746	0.87744	
1221	25009	MA	Essex	11.363419	7.586022	2.489746	0.87744	
1222	25011	MA	Franklin	11.363419	7.586022	2.489746	0.87744	

```
1223  25013    MA     Hampden   11.363419    7.586022   2.489746     0.87744
1224  25015    MA   Hampshire   11.363419    7.586022   2.489746     0.87744
1225  25017    MA   Middlesex   11.363419    7.586022   2.489746     0.87744
1226  25019    MA   Nantucket   11.363419    7.586022   2.489746     0.87744
1227  25021    MA     Norfolk   11.363419    7.586022   2.489746     0.87744
1228  25023    MA    Plymouth   11.363419    7.586022   2.489746     0.87744
1229  25025    MA     Suffolk   11.363419    7.586022   2.489746     0.87744
1230  25027    MA   Worcester   11.363419    7.586022   2.489746     0.87744

      PCT_WIC15
1217   1.666983
1218   1.666983
1219   1.666983
1220   1.666983
1221   1.666983
1222   1.666983
1223   1.666983
1224   1.666983
1225   1.666983
1226   1.666983
1227   1.666983
1228   1.666983
1229   1.666983
1230   1.666983
```

In [23]:
```
prices_taxes_MA = prices_taxes[prices_taxes['State'] == 'MA']
prices_taxes_cols = ['FIPS', 'State', 'County', 'SODATAX_STORES14', 'SODATAX_VENDM14',
prices_taxes_MA = prices_taxes_MA[prices_taxes_cols]
prices_taxes_MA
```

Out[23]:
```
       FIPS State      County  SODATAX_STORES14  SODATAX_VENDM14  \
1217  25001    MA  Barnstable               0.0              0.0
1218  25003    MA   Berkshire               0.0              0.0
1219  25005    MA     Bristol               0.0              0.0
1220  25007    MA       Dukes               0.0              0.0
1221  25009    MA       Essex               0.0              0.0
1222  25011    MA    Franklin               0.0              0.0
1223  25013    MA     Hampden               0.0              0.0
1224  25015    MA   Hampshire               0.0              0.0
1225  25017    MA   Middlesex               0.0              0.0
1226  25019    MA   Nantucket               0.0              0.0
1227  25021    MA     Norfolk               0.0              0.0
1228  25023    MA    Plymouth               0.0              0.0
1229  25025    MA     Suffolk               0.0              0.0
1230  25027    MA   Worcester               0.0              0.0

      CHIPSTAX_STORES14  CHIPSTAX_VENDM14  FOOD_TAX14
1217                0.0               0.0         0.0
```

```
1218                0.0                0.0         0.0
1219                0.0                0.0         0.0
1220                0.0                0.0         0.0
1221                0.0                0.0         0.0
1222                0.0                0.0         0.0
1223                0.0                0.0         0.0
1224                0.0                0.0         0.0
1225                0.0                0.0         0.0
1226                0.0                0.0         0.0
1227                0.0                0.0         0.0
1228                0.0                0.0         0.0
1229                0.0                0.0         0.0
1230                0.0                0.0         0.0
```

In [24]:
```
local_MA = local[local['State'] == 'MA']
local_cols = ['FIPS', 'State', 'County', 'FMRKTPTH16', 'PCT_FMRKT_SNAP16', 'PCT_FMRKT_W
local_MA = local_MA[local_cols]
local_MA
```

Out[24]:

	FIPS	State	County	FMRKTPTH16	PCT_FMRKT_SNAP16	PCT_FMRKT_WIC16	\
1217	25001	MA	Barnstable	0.084004	61.111111	50.000000	
1218	25003	MA	Berkshire	0.149721	36.842105	36.842105	
1219	25005	MA	Bristol	0.044777	36.000000	68.000000	
1220	25007	MA	Dukes	0.173953	0.000000	0.000000	
1221	25009	MA	Essex	0.030808	50.000000	62.500000	
1222	25011	MA	Franklin	0.156290	36.363636	63.636364	
1223	25013	MA	Hampden	0.038423	38.888889	55.555556	
1224	25015	MA	Hampshire	0.080338	53.846154	38.461538	
1225	25017	MA	Middlesex	0.037741	63.333333	68.333333	
1226	25019	MA	Nantucket	0.272529	66.666667	66.666667	
1227	25021	MA	Norfolk	0.031556	50.000000	50.000000	
1228	25023	MA	Plymouth	0.035049	22.222222	22.222222	
1229	25025	MA	Suffolk	0.035704	67.857143	60.714286	
1230	25027	MA	Worcester	0.073207	40.000000	55.000000	

	PCT_FMRKT_WICCASH16	PCT_FMRKT_SFMNP16	PCT_FMRKT_CREDIT16	\
1217	27.777778	55.555556	61.111111	
1218	5.263158	47.368421	73.684211	
1219	24.000000	64.000000	72.000000	
1220	0.000000	0.000000	33.333333	
1221	29.166667	58.333333	75.000000	
1222	45.454545	54.545455	36.363636	
1223	38.888889	55.555556	61.111111	
1224	23.076923	46.153846	53.846154	
1225	18.333333	65.000000	80.000000	
1226	33.333333	66.666667	66.666667	
1227	22.727273	63.636364	63.636364	
1228	27.777778	38.888889	61.111111	

```
1229            28.571429          53.571429          64.285714
1230            13.333333          58.333333          60.000000

      PCT_FMRKT_FRVEG16  PCT_FMRKT_ANMLPROD16  PCT_FMRKT_BAKED16  \
1217          77.777778             77.777778          77.777778
1218          84.210526             84.210526          78.947368
1219          84.000000             60.000000          60.000000
1220           0.000000              0.000000           0.000000
1221          79.166667             62.500000          66.666667
1222          63.636364             63.636364          63.636364
1223          61.111111             50.000000          66.666667
1224          69.230769             69.230769          61.538462
1225          85.000000             76.666667          75.000000
1226          66.666667             33.333333          33.333333
1227          77.272727             68.181818          77.272727
1228          50.000000             44.444444          44.444444
1229          75.000000             64.285714          60.714286
1230          76.666667             75.000000          75.000000

      PCT_FMRKT_OTHERFOOD16  FOODHUB16
1217              77.777778          0
1218              84.210526          1
1219              64.000000          0
1220               0.000000          0
1221              75.000000          0
1222              63.636364          1
1223              61.111111          0
1224              61.538462          0
1225              71.666667          2
1226              33.333333          0
1227              77.272727          1
1228              44.444444          0
1229              64.285714          2
1230              75.000000          0
```

In [25]:
```
health_MA = health[health['State'] == 'MA']
health_cols = ['FIPS', 'State', 'County', 'PCT_DIABETES_ADULTS13', 'PCT_OBESE_ADULTS13'
health_MA = health_MA[health_cols]
health_MA
```

Out[25]:
```
       FIPS State      County  PCT_DIABETES_ADULTS13  PCT_OBESE_ADULTS13  \
1217  25001    MA  Barnstable                    8.6                19.9
1218  25003    MA   Berkshire                    9.7                23.7
1219  25005    MA     Bristol                   11.1                28.3
1220  25007    MA       Dukes                    8.0                22.0
1221  25009    MA       Essex                   10.3                25.5
1222  25011    MA    Franklin                    8.3                21.0
1223  25013    MA     Hampden                   11.1                27.5
```

```
1224  25015    MA   Hampshire                  6.8                18.9
1225  25017    MA   Middlesex                  8.3                22.6
1226  25019    MA   Nantucket                  8.1                20.0
1227  25021    MA     Norfolk                  8.6                20.5
1228  25023    MA    Plymouth                 10.7                27.8
1229  25025    MA     Suffolk                  7.5                21.3
1230  25027    MA   Worcester                  9.7                27.0

      PCT_HSPA15  RECFACPTH14
1217        24.1     0.223345
1218        24.1     0.108767
1219        24.1     0.138941
1220        24.1     0.403319
1221        24.1     0.152128
1222        24.1     0.127007
1223        24.1     0.085441
1224        24.1     0.118057
1225        24.1     0.206328
1226        24.1     0.276345
1227        24.1     0.189237
1228        24.1     0.167646
1229        24.1     0.143368
1230        24.1     0.129076
```

In [26]:
```
socioeconomic_MA = socioeconomic[socioeconomic['State'] == 'MA']
socioeconomic_cols = ['FIPS', 'State', 'County', 'PCT_NHWHITE10', 'PCT_NHBLACK10', 'PCT
socioeconomic_MA = socioeconomic_MA[socioeconomic_cols]
socioeconomic_MA
```

Out[26]:
```
       FIPS State      County  PCT_NHWHITE10  PCT_NHBLACK10  PCT_HISP10  \
1217  25001    MA  Barnstable      91.402486       1.786575    2.171033
1218  25003    MA   Berkshire      90.631692       2.540791    3.452244
1219  25005    MA     Bristol      85.602196       2.870040    6.022415
1220  25007    MA       Dukes      86.332023       2.884790    2.322347
1221  25009    MA       Essex      76.031509       2.632815   16.516654
1222  25011    MA    Franklin      92.442414       0.984980    3.152497
1223  25013    MA     Hampden      67.713651       7.700706   20.879846
1224  25015    MA   Hampshire      86.189904       2.243801    4.715967
1225  25017    MA   Middlesex      76.526012       4.373206    6.543209
1226  25019    MA   Nantucket      80.534801       6.527723    9.408179
1227  25021    MA     Norfolk      80.309756       5.415369    3.280018
1228  25023    MA    Plymouth      83.921005       6.867993    3.155870
1229  25025    MA     Suffolk      48.056502      19.802693   19.868481
1230  25027    MA   Worcester      80.683412       3.642092    9.444845

      PCT_NHASIAN10  PCT_NHNA10  PCT_NHPI10  PCT_65OLDER10  PCT_18YOUNGER10  \
1217       1.048229    0.548433    0.030108      24.956922        17.253854
1218       1.220860    0.150131    0.017528      18.584199        19.527660
```

```
1219       1.849403    0.198619    0.021704      14.204109        22.325798
1220       0.743877    1.058361    0.024191      16.322951        19.189598
1221       3.076596    0.123123    0.021664      14.140043        23.156417
1222       1.242784    0.245194    0.012610      15.241271        19.710811
1223       1.928197    0.154696    0.024596      14.184772        23.708171
1224       4.505946    0.153087    0.025936      12.665739        16.931933
1225       9.269270    0.104984    0.022021      13.107376        21.318754
1226       1.160047    0.058985    0.009831      12.062525        20.723555
1227       8.589849    0.120891    0.015950      14.504584        22.677499
1228       1.197772    0.206903    0.023842      13.910559        24.140314
1229       8.166360    0.189329    0.027146      10.488032        17.489055
1230       3.956787    0.162544    0.022290      12.777502        23.446313

      MEDHHINC15  POVRATE15  METRO13
1217     65735.0        7.6        1
1218     50646.0       14.3        1
1219     59839.0       12.6        1
1220     64456.0        8.5        0
1221     68237.0       11.5        1
1222     57325.0       11.8        0
1223     51415.0       17.1        1
1224     60853.0       15.3        1
1225     90025.0        7.6        1
1226     86014.0        7.3        0
1227     93187.0        7.1        1
1228     74736.0        9.7        1
1229     56530.0       19.8        1
1230     65621.0       12.1        1
```

In [27]:
```
dfs_MA = [stores_MA, assistance_MA, local_MA, health_MA, restaurants_MA, socioeconomic_
for df_MA in dfs_MA:
    df_MA.drop(columns=['State', 'County'], axis=1, inplace=True)
    df_MA.set_index('FIPS', inplace=True)

# Then, we'll also set the index on the Access df. This will be the dataframe we join t
access_MA.set_index('FIPS', inplace=True)
master_df_MA = access_MA.join(dfs_MA)
```

In [28]:
```
master_df_MA.shape
```

Out[28]:
```
(14, 58)
```

In [29]:
```
master_df_MA.isnull().sum()
master_df_MA = master_df_MA.dropna()
master_df_MA.isnull().sum()
```

Out[29]:
```
State                   0
County                  0
PCT_LACCESS_POP15       0
```

PCT_LACCESS_LOWI15	0
PCT_LACCESS_HHNV15	0
PCT_LACCESS_SNAP15	0
PCT_LACCESS_CHILD15	0
PCT_LACCESS_SENIORS15	0
PCT_LACCESS_WHITE15	0
PCT_LACCESS_BLACK15	0
PCT_LACCESS_HISP15	0
PCT_LACCESS_NHASIAN15	0
PCT_LACCESS_NHNA15	0
PCT_LACCESS_NHPI15	0
PCT_LACCESS_MULTIR15	0
GROCPTH14	0
SUPERCPTH14	0
CONVSPTH14	0
SPECSPTH14	0
SNAPSPTH16	0
PCT_SNAP16	0
PCT_NSLP15	0
PCT_SBP15	0
PCT_SFSP15	0
PCT_WIC15	0
FMRKTPTH16	0
PCT_FMRKT_SNAP16	0
PCT_FMRKT_WIC16	0
PCT_FMRKT_WICCASH16	0
PCT_FMRKT_SFMNP16	0
PCT_FMRKT_CREDIT16	0
PCT_FMRKT_FRVEG16	0
PCT_FMRKT_ANMLPROD16	0
PCT_FMRKT_BAKED16	0
PCT_FMRKT_OTHERFOOD16	0
FOODHUB16	0
PCT_DIABETES_ADULTS13	0
PCT_OBESE_ADULTS13	0
PCT_HSPA15	0
RECFACPTH14	0
FFRPTH14	0
FSRPTH14	0
PCT_NHWHITE10	0
PCT_NHBLACK10	0
PCT_HISP10	0
PCT_NHASIAN10	0
PCT_NHNA10	0
PCT_NHPI10	0
PCT_65OLDER10	0
PCT_18YOUNGER10	0
MEDHHINC15	0

```
POVRATE15                0
METRO13                  0
SODATAX_STORES14         0
SODATAX_VENDM14          0
CHIPSTAX_STORES14        0
CHIPSTAX_VENDM14         0
FOOD_TAX14               0
dtype: int64
```

```
In [30]: import seaborn as sns
         from sklearn import linear_model
         from sklearn.metrics import r2_score
         from sklearn.model_selection import train_test_split
```

```
In [31]:
```

```
Out[31]: (14, 58)
```

```
In [50]: features_new = stores_cols + assistance_cols + local_cols + health_cols + restaurant_co
         prediction_features = list(set(features_new) - set(['FIPS', 'State', 'County', 'PCT_DIA
         target = 'PCT_OBESE_ADULTS13'
```

```
In [51]: # Split the data into train and test sets
         X_train, X_test, y_train, y_test = train_test_split(master_df_MA[prediction_features],

         # Create linear regression object
         regr = linear_model.LinearRegression()

         # Train our model
         regr.fit(X_train, y_train)

         # Make predictions
         y_pred = regr.predict(X_test)

         # R2 scores
         print(r2_score(y_train, regr.predict(X_train)))
         print(r2_score(y_test, y_pred))

         # overfitted way too much
```

```
1.0
0.13681961479862303


/usr/local/lib/python2.7/site-packages/sklearn/linear_model/base.py:509: RuntimeWarning: interna
  linalg.lstsq(X, y)
```

```
In [52]: master_df_MA
```

Out[52]:

FIPS	State	County	PCT_LACCESS_POP15	PCT_LACCESS_LOWI15
25001	MA	Barnstable	48.379442	9.717726
25003	MA	Berkshire	24.321984	5.971745
25005	MA	Bristol	28.955995	4.502100
25007	MA	Dukes	7.120600	1.372931
25009	MA	Essex	29.959103	4.413228
25011	MA	Franklin	15.492427	5.682865
25013	MA	Hampden	22.289413	4.932812
25015	MA	Hampshire	38.655972	9.980293
25017	MA	Middlesex	24.825819	2.993366
25019	MA	Nantucket	10.422783	1.150662
25021	MA	Norfolk	31.694543	3.476869
25023	MA	Plymouth	40.820126	5.824657
25025	MA	Suffolk	0.750733	0.211504
25027	MA	Worcester	28.365209	5.020362

FIPS	PCT_LACCESS_HHNV15	PCT_LACCESS_SNAP15	PCT_LACCESS_CHILD15
25001	2.312366	3.192731	8.717733
25003	2.486676	2.623480	4.601347
25005	1.451063	2.184273	6.523161
25007	1.065238	0.137535	1.459410
25009	1.520608	1.926098	7.005516
25011	2.958759	2.178276	3.040781
25013	1.822251	2.422411	4.661253
25015	3.888760	2.737564	5.955018
25017	1.087936	1.062284	6.147123
25019	1.222215	0.301460	2.461146
25021	1.551223	1.520745	8.180235
25023	1.814911	2.746281	10.072809
25025	0.146702	0.126259	0.144534
25027	2.014543	2.523371	6.735784

FIPS	PCT_LACCESS_SENIORS15	PCT_LACCESS_WHITE15	PCT_LACCESS_BLACK15
25001	11.469996	45.383442	0.705073
25003	4.731523	22.841151	0.482051
25005	3.940878	27.101606	0.569744
25007	1.248538	6.207773	0.155550
25009	4.312460	27.287571	0.513947
25011	2.103533	14.261431	0.294346
25013	3.765549	19.934845	0.949074
25015	4.657704	33.119500	1.242268
25017	3.336203	21.883684	0.465024
25019	0.715661	8.843923	1.051898
25021	4.323727	28.006690	1.306877
25023	5.695550	38.532252	0.735054

25025		0.116408	0.547030		0.098940	
25027		3.808896	25.709256		0.680676	

	...	PCT_65OLDER10	PCT_18YOUNGER10	MEDHHINC15	POVRATE15	\
FIPS	...					
25001	...	24.956922	17.253854	65735.0	7.6	
25003	...	18.584199	19.527660	50646.0	14.3	
25005	...	14.204109	22.325798	59839.0	12.6	
25007	...	16.322951	19.189598	64456.0	8.5	
25009	...	14.140043	23.156417	68237.0	11.5	
25011	...	15.241271	19.710811	57325.0	11.8	
25013	...	14.184772	23.708171	51415.0	17.1	
25015	...	12.665739	16.931933	60853.0	15.3	
25017	...	13.107376	21.318754	90025.0	7.6	
25019	...	12.062525	20.723555	86014.0	7.3	
25021	...	14.504584	22.677499	93187.0	7.1	
25023	...	13.910559	24.140314	74736.0	9.7	
25025	...	10.488032	17.489055	56530.0	19.8	
25027	...	12.777502	23.446313	65621.0	12.1	

	METRO13	SODATAX_STORES14	SODATAX_VENDM14	CHIPSTAX_STORES14	\
FIPS					
25001	1	0.0	0.0	0.0	
25003	1	0.0	0.0	0.0	
25005	1	0.0	0.0	0.0	
25007	0	0.0	0.0	0.0	
25009	1	0.0	0.0	0.0	
25011	0	0.0	0.0	0.0	
25013	1	0.0	0.0	0.0	
25015	1	0.0	0.0	0.0	
25017	1	0.0	0.0	0.0	
25019	0	0.0	0.0	0.0	
25021	1	0.0	0.0	0.0	
25023	1	0.0	0.0	0.0	
25025	1	0.0	0.0	0.0	
25027	1	0.0	0.0	0.0	

	CHIPSTAX_VENDM14	FOOD_TAX14
FIPS		
25001	0.0	0.0
25003	0.0	0.0
25005	0.0	0.0
25007	0.0	0.0
25009	0.0	0.0
25011	0.0	0.0
25013	0.0	0.0
25015	0.0	0.0
25017	0.0	0.0

```
25019              0.0         0.0
25021              0.0         0.0
25023              0.0         0.0
25025              0.0         0.0
25027              0.0         0.0

[14 rows x 58 columns]
```

In [54]:
```
#testing all counties in country for more EDA
```

Out[54]:
```
Index([u'Food Environment Atlas data download'], dtype='object')
```

In [ ]:

In [ ]:

In [55]:
```
# We're going to join all these sheets, so we'll drop redundant state and county cols f
# We'll also set the FIPS ID col to be index so we can use that for the join.
dfs = [stores, assistance, insecurity, local, health, restaurants, socioeconomic, price
for df in dfs:
    df.drop(columns=['State', 'County'], axis=1, inplace=True)
    df.set_index('FIPS', inplace=True)

# Then, we'll also set the index on the Access df. This will be the dataframe we join t
access.set_index('FIPS', inplace=True)
```

In [56]:
```
# Combine all sheets into one dataframe by joining on FIPS col.
# Now we have a master dataframe containing the cols from all sheets.
master_df = access.join(dfs)
```

In [66]:
```
df_cols = list(set(features_new) - set(['State', 'County', 'FIPS']))

# Create a dataframe with just the cols of interest.
df = master_df[df_cols]
df.shape
```

Out[66]:
```
(3143, 56)
```

In [67]:
```
# Check for null values...
df.isnull().sum()
```

Out[67]:
```
PCT_OBESE_ADULTS13          1
PCT_SNAP16                  0
PCT_FMRKT_OTHERFOOD16     895
PCT_DIABETES_ADULTS13       1
PCT_SFSP15                  0
PCT_NHNA10                  0
PCT_SBP15                   0
SODATAX_STORES14            0
```

CHIPSTAX_VENDM14	0
MEDHHINC15	4
PCT_NHWHITE10	0
FFRPTH14	0
FSRPTH14	0
PCT_FMRKT_WIC16	895
METRO13	0
GROCPTH14	0
PCT_LACCESS_NHPI15	19
RECFACPTH14	0
PCT_HISP10	0
PCT_FMRKT_BAKED16	895
PCT_LACCESS_SNAP15	20
PCT_FMRKT_WICCASH16	895
SODATAX_VENDM14	0
PCT_LACCESS_NHASIAN15	19
SNAPSPTH16	29
PCT_WIC15	0
PCT_LACCESS_SENIORS15	19
FMRKTPTH16	2
POVRATE15	4
SUPERCPTH14	0
PCT_FMRKT_FRVEG16	895
PCT_HSPA15	1118
PCT_FMRKT_SNAP16	895
PCT_65OLDER10	0
CHIPSTAX_STORES14	0
PCT_NSLP15	0
PCT_18YOUNGER10	0
PCT_LACCESS_LOWI15	20
PCT_LACCESS_NHNA15	19
CONVSPTH14	0
FOODHUB16	0
PCT_LACCESS_WHITE15	19
PCT_FMRKT_ANMLPROD16	895
PCT_LACCESS_MULTIR15	19
PCT_LACCESS_CHILD15	19
PCT_NHBLACK10	0
PCT_LACCESS_HHNV15	3
PCT_FMRKT_CREDIT16	895
FOOD_TAX14	0
PCT_NHPI10	0
SPECSPTH14	0
PCT_NHASIAN10	0
PCT_FMRKT_SFMNP16	895
PCT_LACCESS_HISP15	19
PCT_LACCESS_POP15	19
PCT_LACCESS_BLACK15	19

```
dtype: int64
```

In [68]:
```
# ...and drop them
df = df.dropna()
df.isnull().sum()
```

Out[68]:
```
PCT_OBESE_ADULTS13       0
PCT_SNAP16               0
PCT_FMRKT_OTHERFOOD16    0
PCT_DIABETES_ADULTS13    0
PCT_SFSP15               0
PCT_NHNA10               0
PCT_SBP15                0
SODATAX_STORES14         0
CHIPSTAX_VENDM14         0
MEDHHINC15               0
PCT_NHWHITE10            0
FFRPTH14                 0
FSRPTH14                 0
PCT_FMRKT_WIC16          0
METRO13                  0
GROCPTH14                0
PCT_LACCESS_NHPI15       0
RECFACPTH14              0
PCT_HISP10               0
PCT_FMRKT_BAKED16        0
PCT_LACCESS_SNAP15       0
PCT_FMRKT_WICCASH16      0
SODATAX_VENDM14          0
PCT_LACCESS_NHASIAN15    0
SNAPSPTH16               0
PCT_WIC15                0
PCT_LACCESS_SENIORS15    0
FMRKTPTH16               0
POVRATE15                0
SUPERCPTH14              0
PCT_FMRKT_FRVEG16        0
PCT_HSPA15               0
PCT_FMRKT_SNAP16         0
PCT_65OLDER10            0
CHIPSTAX_STORES14        0
PCT_NSLP15               0
PCT_18YOUNGER10          0
PCT_LACCESS_LOWI15       0
PCT_LACCESS_NHNA15       0
CONVSPTH14               0
FOODHUB16                0
PCT_LACCESS_WHITE15      0
```

```
PCT_FMRKT_ANMLPROD16     0
PCT_LACCESS_MULTIR15     0
PCT_LACCESS_CHILD15      0
PCT_NHBLACK10            0
PCT_LACCESS_HHNV15       0
PCT_FMRKT_CREDIT16       0
FOOD_TAX14               0
PCT_NHPI10               0
SPECSPTH14               0
PCT_NHASIAN10            0
PCT_FMRKT_SFMNP16        0
PCT_LACCESS_HISP15       0
PCT_LACCESS_POP15        0
PCT_LACCESS_BLACK15      0
dtype: int64
```

```
In [60]: # dtypes look good
         print(df.dtypes)
```

```
PCT_SNAP16               float64
PCT_FMRKT_OTHERFOOD16    float64
PCT_SFSP15               float64
PCT_NHNA10               float64
PCT_SBP15                float64
SODATAX_STORES14         float64
CHIPSTAX_VENDM14         float64
MEDHHINC15               float64
PCT_NHWHITE10            float64
FFRPTH14                 float64
FSRPTH14                 float64
PCT_FMRKT_WIC16          float64
METRO13                    int64
GROCPTH14                float64
PCT_LACCESS_NHPI15       float64
RECFACPTH14              float64
PCT_HISP10               float64
PCT_FMRKT_BAKED16        float64
PCT_LACCESS_SNAP15       float64
PCT_FMRKT_WICCASH16      float64
SODATAX_VENDM14          float64
PCT_LACCESS_NHASIAN15    float64
SNAPSPTH16               float64
PCT_WIC15                float64
PCT_LACCESS_SENIORS15    float64
FMRKTPTH16               float64
POVRATE15                float64
SUPERCPTH14              float64
PCT_FMRKT_FRVEG16        float64
```

```
PCT_HSPA15               float64
PCT_FMRKT_SNAP16         float64
PCT_65OLDER10            float64
CHIPSTAX_STORES14        float64
PCT_NSLP15               float64
PCT_18YOUNGER10          float64
PCT_LACCESS_LOWI15       float64
PCT_LACCESS_NHNA15       float64
CONVSPTH14               float64
FOODHUB16                  int64
PCT_LACCESS_WHITE15      float64
PCT_FMRKT_ANMLPROD16     float64
PCT_LACCESS_MULTIR15     float64
PCT_LACCESS_CHILD15      float64
PCT_NHBLACK10            float64
PCT_LACCESS_HHNV15       float64
PCT_FMRKT_CREDIT16       float64
FOOD_TAX14               float64
PCT_NHPI10               float64
SPECSPTH14               float64
PCT_NHASIAN10            float64
PCT_FMRKT_SFMNP16        float64
PCT_LACCESS_HISP15       float64
PCT_LACCESS_POP15        float64
PCT_LACCESS_BLACK15      float64
dtype: object
```

In [69]:
```
# Split the data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(df[prediction_features], df[target]

# Create linear regression object
regr = linear_model.LinearRegression()

# Train our model
regr.fit(X_train, y_train)

# Make predictions
y_pred = regr.predict(X_test)

# R2 scores
print(r2_score(y_train, regr.predict(X_train)))
print(r2_score(y_test, y_pred))
```

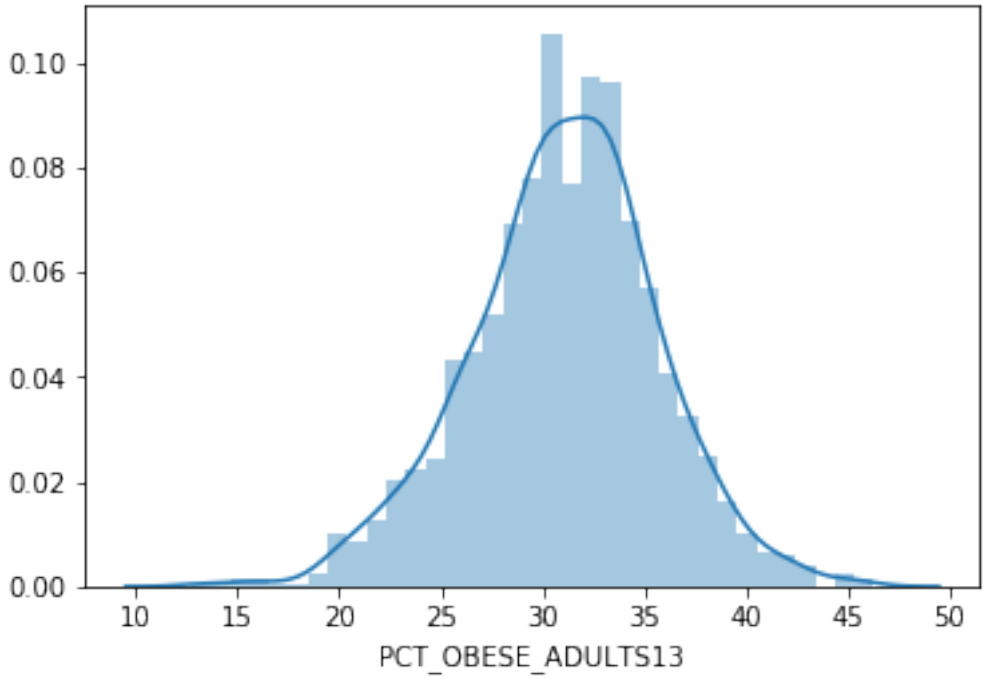
```
0.7004361198768504
0.6217465320137188
```

In [70]:
```
#more EDA
```

```
sns.distplot(df['PCT_OBESE_ADULTS13'])
```

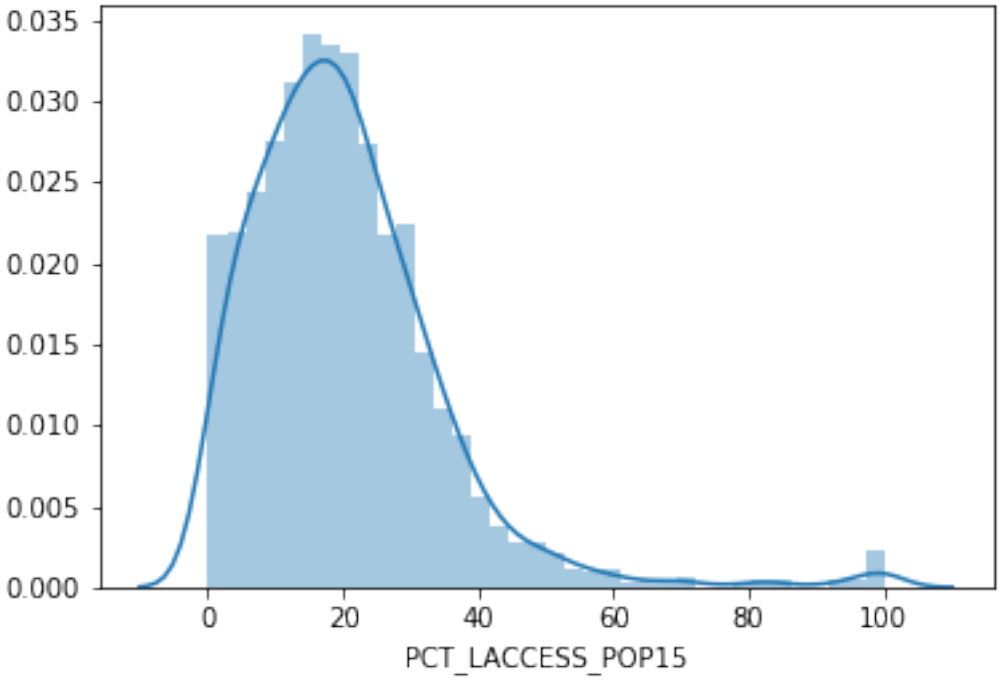
```
/usr/local/lib/python2.7/site-packages/scipy/stats/stats.py:1713: FutureWarning: Using a non-tup
  return np.add.reduce(sorted[indexer] * weights, axis=axis) / sumval
```

Out[70]: <matplotlib.axes._subplots.AxesSubplot at 0x11433b410>

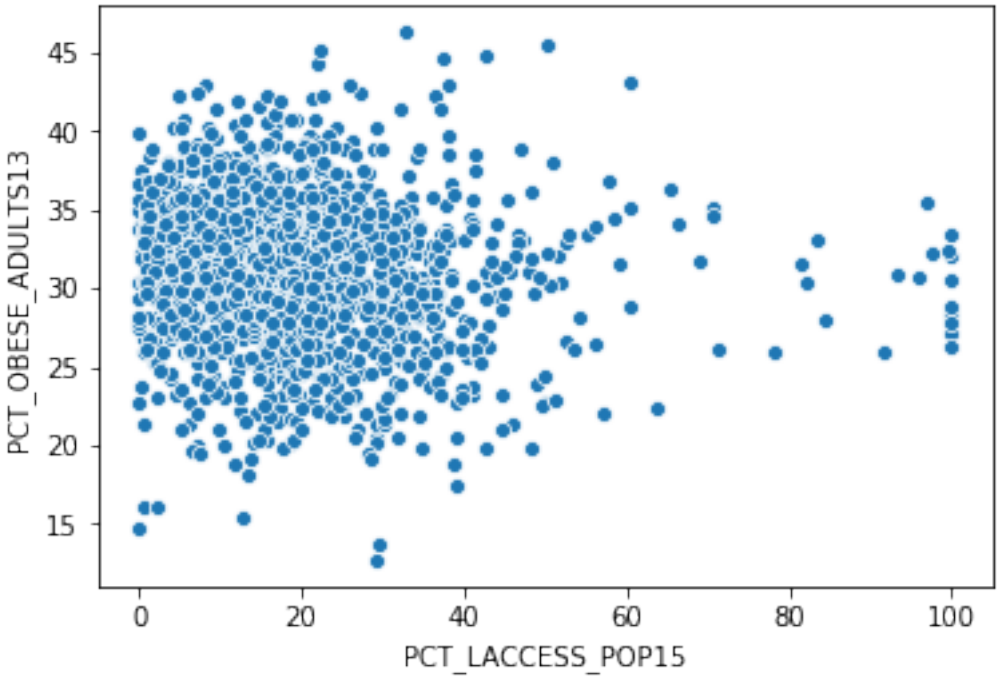


In [71]:
```
sns.distplot(df['PCT_LACCESS_POP15'])
```

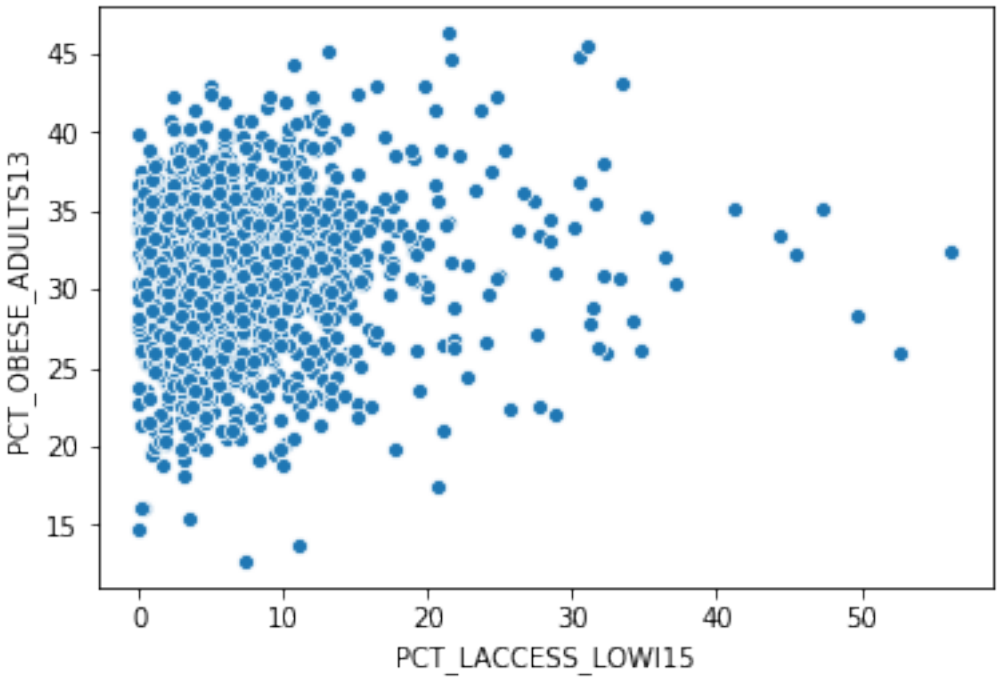
Out[71]: <matplotlib.axes._subplots.AxesSubplot at 0x111432b10>



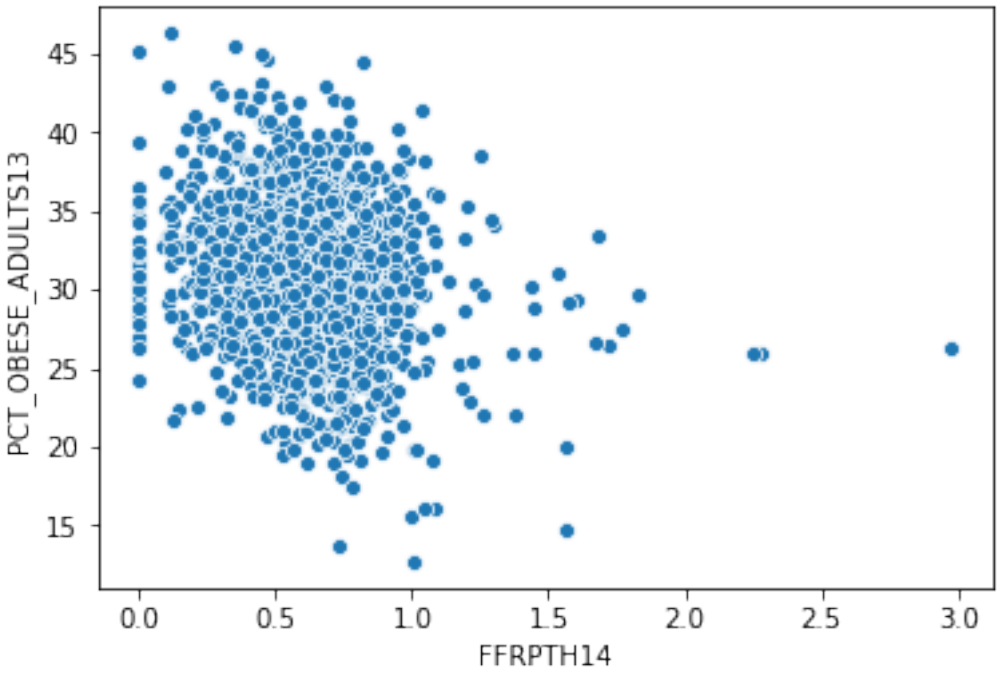

```
In [72]: plot = sns.scatterplot(x="PCT_LACCESS_POP15", y="PCT_OBESE_ADULTS13", data=df)
```



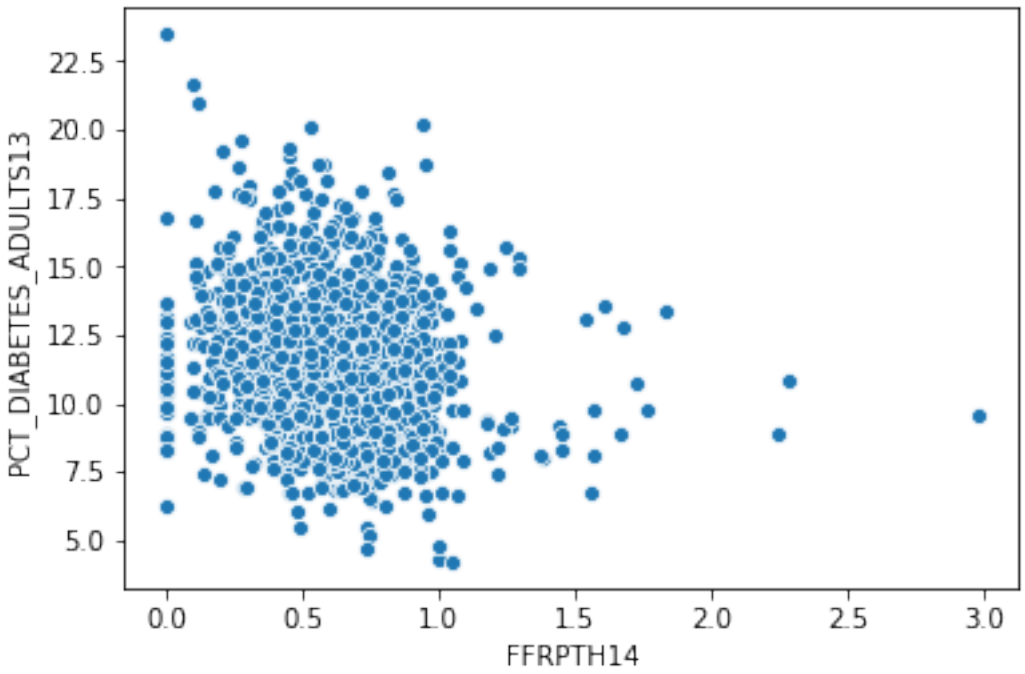
```
In [73]: plot = sns.scatterplot(x="PCT_LACCESS_LOWI15", y="PCT_OBESE_ADULTS13", data=df)
```



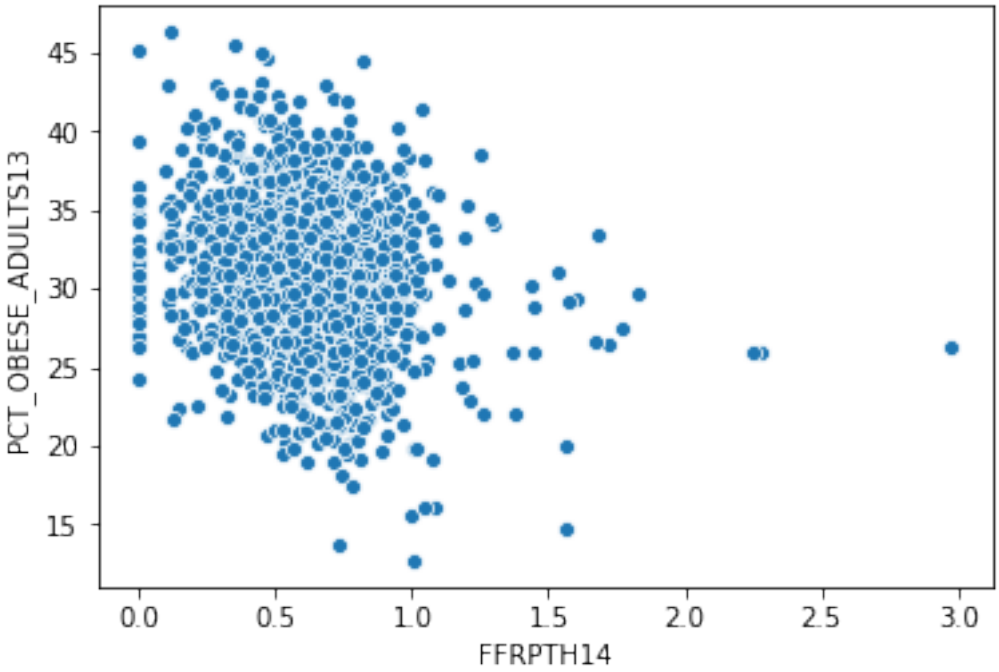

```
In [76]: plot = sns.scatterplot(x="FFRPTH14", y="PCT_OBESE_ADULTS13", data=df)
```



In [77]:
```
plot = sns.scatterplot(x="FFRPTH14", y="PCT_DIABETES_ADULTS13", data=df)
```



In [81]:
```
# obesity rate vs fast food
plot = sns.scatterplot(x="FFRPTH14", y="PCT_OBESE_ADULTS13", data=df)
```



In [ ]:

In [ ]:

In [ ]:

In [ ]:

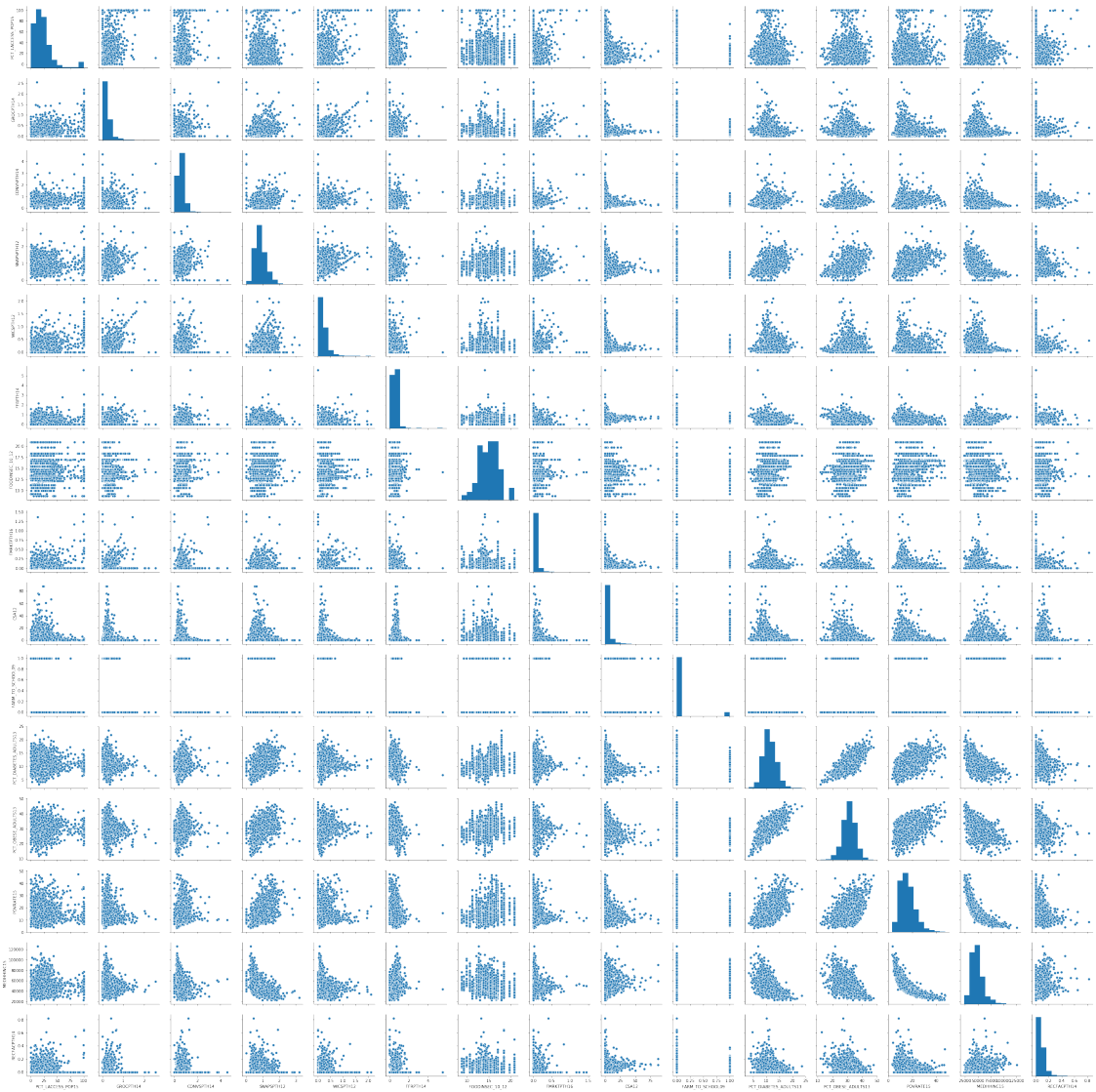
In [ ]:

In [ ]:

In [148]:
```
import seaborn as sns
```

In [168]:
```
# We can look at scatter plots of each feature in pairs, to see if there is a correlat
# You can double-click on a plot to enlarge.
# We may need to back-track and look at some other variables.
# This is the part where it would be good to get some input from the TAs.
sns.pairplot(df)
```

Out[168]: <seaborn.axisgrid.PairGrid at 0x153c80668>



```
In [153]: from sklearn import linear_model
          from sklearn.metrics import r2_score
          from sklearn.model_selection import train_test_split

In [164]: features = ['PCT_LACCESS_POP15', 'GROCPTH14', 'CONVSPTH14', 'SNAPSPTH12', 'WICSPTH12',
                      'FOODINSEC_10_12', 'FMRKTPTH16', 'CSA12', 'FARM_TO_SCHOOL09', 'POVRATE15',
                      'PCT_DIABETES_ADULTS13', 'RECFACPTH14']
          target = ['PCT_OBESE_ADULTS13']

In [167]: # Split the data into train and test sets
          X_train, X_test, y_train, y_test = train_test_split(df[features], df[target], test_siz

          # Create linear regression object
```

```
        regr = linear_model.LinearRegression()

        # Train our model
        regr.fit(X_train, y_train)

        # Make predictions
        y_pred = regr.predict(X_test)

        # R2 scores
        print(r2_score(y_train, regr.predict(X_train)))
        print(r2_score(y_test, y_pred))
```

```
0.5452410535840758
0.5076118584156646
```