Real Estate

June 22, 2023

[92]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

[93]:
```
test=pd.read_csv('test.csv')
train=pd.read_csv('train.csv')
```

[94]:
```
#shape of data
print(test.shape)
print(train.shape)
```

```
(11709, 80)
(27321, 80)
```

[95]:
```
train.head()
```

[95]:
```
      UID  BLOCKID  SUMLEVEL  COUNTYID  STATEID        state state_ab  \
0  267822      NaN       140        53       36     New York       NY
1  246444      NaN       140       141       18      Indiana       IN
2  245683      NaN       140        63       18      Indiana       IN
3  279653      NaN       140       127       72  Puerto Rico       PR
4  247218      NaN       140       161       20       Kansas       KS

         city           place   type  … female_age_mean  female_age_median  \
0    Hamilton        Hamilton   City  …        44.48629           45.33333
1  South Bend        Roseland   City  …        36.48391           37.58333
2    Danville        Danville   City  …        42.15810           42.83333
3    San Juan        Guaynabo  Urban  …        47.77526           50.58333
4   Manhattan  Manhattan City   City  …        24.17693           21.58333

   female_age_stdev  female_age_sample_weight  female_age_samples  pct_own  \
0          22.51276                 685.33845              2618.0  0.79046
1          23.43353                 267.23367              1284.0  0.52483
2          23.94119                 707.01963              3238.0  0.85331
3          24.32015                 362.20193              1559.0  0.65037
4          11.10484                1854.48652              3051.0  0.13046
```

```
   married  married_snp  separated  divorced
0  0.57851      0.01882    0.01240   0.08770
1  0.34886      0.01426    0.01426   0.09030
2  0.64745      0.02830    0.01607   0.10657
3  0.47257      0.02021    0.02021   0.10106
4  0.12356      0.00000    0.00000   0.03109

[5 rows x 80 columns]
```

[96]:
```
train.columns
```

[96]:
```
Index(['UID', 'BLOCKID', 'SUMLEVEL', 'COUNTYID', 'STATEID', 'state',
       'state_ab', 'city', 'place', 'type', 'primary', 'zip_code', 'area_code',
       'lat', 'lng', 'ALand', 'AWater', 'pop', 'male_pop', 'female_pop',
       'rent_mean', 'rent_median', 'rent_stdev', 'rent_sample_weight',
       'rent_samples', 'rent_gt_10', 'rent_gt_15', 'rent_gt_20', 'rent_gt_25',
       'rent_gt_30', 'rent_gt_35', 'rent_gt_40', 'rent_gt_50',
       'universe_samples', 'used_samples', 'hi_mean', 'hi_median', 'hi_stdev',
       'hi_sample_weight', 'hi_samples', 'family_mean', 'family_median',
       'family_stdev', 'family_sample_weight', 'family_samples',
       'hc_mortgage_mean', 'hc_mortgage_median', 'hc_mortgage_stdev',
       'hc_mortgage_sample_weight', 'hc_mortgage_samples', 'hc_mean',
       'hc_median', 'hc_stdev', 'hc_samples', 'hc_sample_weight',
       'home_equity_second_mortgage', 'second_mortgage', 'home_equity', 'debt',
       'second_mortgage_cdf', 'home_equity_cdf', 'debt_cdf', 'hs_degree',
       'hs_degree_male', 'hs_degree_female', 'male_age_mean',
       'male_age_median', 'male_age_stdev', 'male_age_sample_weight',
       'male_age_samples', 'female_age_mean', 'female_age_median',
       'female_age_stdev', 'female_age_sample_weight', 'female_age_samples',
       'pct_own', 'married', 'married_snp', 'separated', 'divorced'],
      dtype='object')
```

[97]:
```
train.describe()
```

[97]:
```
                 UID  BLOCKID  SUMLEVEL      COUNTYID       STATEID  \
count   27321.000000      0.0   27321.0  27321.000000  27321.000000
mean   257331.996303      NaN     140.0     85.646426     28.271806
std     21343.859725      NaN       0.0     98.333097     16.392846
min    220342.000000      NaN     140.0      1.000000      1.000000
25%    238816.000000      NaN     140.0     29.000000     13.000000
50%    257220.000000      NaN     140.0     63.000000     28.000000
75%    275818.000000      NaN     140.0    109.000000     42.000000
max    294334.000000      NaN     140.0    840.000000     72.000000

           zip_code     area_code           lat           lng         ALand  \
count  27321.000000  27321.000000  27321.000000  27321.000000  2.732100e+04
```

```
mean   50081.999524    596.507668     37.508813   -91.288394  1.295106e+08
std    29558.115660    232.497482      5.588268    16.343816  1.275531e+09
min      602.000000    201.000000     17.929085  -165.453872  4.113400e+04
25%    26554.000000    405.000000     33.899064   -97.816067  1.799408e+06
50%    47715.000000    614.000000     38.755183   -86.554374  4.866940e+06
75%    77093.000000    801.000000     41.380606   -79.782503  3.359820e+07
max    99925.000000    989.000000     67.074018   -65.379332  1.039510e+11

       …  female_age_mean  female_age_median  female_age_stdev  \
count  …     27115.000000       27115.000000      27115.000000
mean   …        40.319803          40.355099         22.178745
std    …         5.886317           8.039585          2.540257
min    …        16.008330          13.250000          0.556780
25%    …        36.892050          34.916670         21.312135
50%    …        40.373320          40.583330         22.514410
75%    …        43.567120          45.416670         23.575260
max    …        79.837390          82.250000         30.241270

       female_age_sample_weight  female_age_samples       pct_own  \
count              27115.000000        27115.000000  27053.000000
mean                 544.238432         2208.761903      0.640434
std                  283.546896         1089.316999      0.226640
min                    0.664700            2.000000      0.000000
25%                  355.995825         1471.000000      0.502780
50%                  503.643890         2066.000000      0.690840
75%                  680.275055         2772.000000      0.817460
max                 6197.995200        27250.000000      1.000000

            married   married_snp     separated      divorced
count  27130.000000  27130.000000  27130.000000  27130.000000
mean       0.508300      0.047537      0.019089      0.100248
std        0.136860      0.037640      0.020796      0.049055
min        0.000000      0.000000      0.000000      0.000000
25%        0.425102      0.020810      0.004530      0.065800
50%        0.526665      0.038840      0.013460      0.095205
75%        0.605760      0.065100      0.027487      0.129000
max        1.000000      0.714290      0.714290      1.000000

[8 rows x 74 columns]
```

[98]:
```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27321 entries, 0 to 27320
Data columns (total 80 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
```

```
0   UID                    27321 non-null  int64
1   BLOCKID                0 non-null      float64
2   SUMLEVEL               27321 non-null  int64
3   COUNTYID               27321 non-null  int64
4   STATEID                27321 non-null  int64
5   state                  27321 non-null  object
6   state_ab               27321 non-null  object
7   city                   27321 non-null  object
8   place                  27321 non-null  object
9   type                   27321 non-null  object
10  primary                27321 non-null  object
11  zip_code               27321 non-null  int64
12  area_code              27321 non-null  int64
13  lat                    27321 non-null  float64
14  lng                    27321 non-null  float64
15  ALand                  27321 non-null  float64
16  AWater                 27321 non-null  int64
17  pop                    27321 non-null  int64
18  male_pop               27321 non-null  int64
19  female_pop             27321 non-null  int64
20  rent_mean              27007 non-null  float64
21  rent_median            27007 non-null  float64
22  rent_stdev             27007 non-null  float64
23  rent_sample_weight     27007 non-null  float64
24  rent_samples           27007 non-null  float64
25  rent_gt_10             27007 non-null  float64
26  rent_gt_15             27007 non-null  float64
27  rent_gt_20             27007 non-null  float64
28  rent_gt_25             27007 non-null  float64
29  rent_gt_30             27007 non-null  float64
30  rent_gt_35             27007 non-null  float64
31  rent_gt_40             27007 non-null  float64
32  rent_gt_50             27007 non-null  float64
33  universe_samples       27321 non-null  int64
34  used_samples           27321 non-null  int64
35  hi_mean                27053 non-null  float64
36  hi_median              27053 non-null  float64
37  hi_stdev               27053 non-null  float64
38  hi_sample_weight       27053 non-null  float64
39  hi_samples             27053 non-null  float64
40  family_mean            27023 non-null  float64
41  family_median          27023 non-null  float64
42  family_stdev           27023 non-null  float64
43  family_sample_weight   27023 non-null  float64
44  family_samples         27023 non-null  float64
45  hc_mortgage_mean       26748 non-null  float64
46  hc_mortgage_median     26748 non-null  float64
47  hc_mortgage_stdev      26748 non-null  float64
```

```
 48  hc_mortgage_sample_weight    26748 non-null  float64
 49  hc_mortgage_samples          26748 non-null  float64
 50  hc_mean                      26721 non-null  float64
 51  hc_median                    26721 non-null  float64
 52  hc_stdev                     26721 non-null  float64
 53  hc_samples                   26721 non-null  float64
 54  hc_sample_weight             26721 non-null  float64
 55  home_equity_second_mortgage  26864 non-null  float64
 56  second_mortgage              26864 non-null  float64
 57  home_equity                  26864 non-null  float64
 58  debt                         26864 non-null  float64
 59  second_mortgage_cdf          26864 non-null  float64
 60  home_equity_cdf              26864 non-null  float64
 61  debt_cdf                     26864 non-null  float64
 62  hs_degree                    27131 non-null  float64
 63  hs_degree_male               27121 non-null  float64
 64  hs_degree_female             27098 non-null  float64
 65  male_age_mean                27132 non-null  float64
 66  male_age_median              27132 non-null  float64
 67  male_age_stdev               27132 non-null  float64
 68  male_age_sample_weight       27132 non-null  float64
 69  male_age_samples             27132 non-null  float64
 70  female_age_mean              27115 non-null  float64
 71  female_age_median            27115 non-null  float64
 72  female_age_stdev             27115 non-null  float64
 73  female_age_sample_weight     27115 non-null  float64
 74  female_age_samples           27115 non-null  float64
 75  pct_own                      27053 non-null  float64
 76  married                      27130 non-null  float64
 77  married_snp                  27130 non-null  float64
 78  separated                    27130 non-null  float64
 79  divorced                     27130 non-null  float64
dtypes: float64(62), int64(12), object(6)
memory usage: 16.7+ MB
```

[99]:
```
test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11709 entries, 0 to 11708
Data columns (total 80 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   UID                          11709 non-null  int64
 1   BLOCKID                      0 non-null      float64
 2   SUMLEVEL                     11709 non-null  int64
 3   COUNTYID                     11709 non-null  int64
 4   STATEID                      11709 non-null  int64
 5   state                        11709 non-null  object
```

```
6   state_ab                     11709 non-null  object
7   city                         11709 non-null  object
8   place                        11709 non-null  object
9   type                         11709 non-null  object
10  primary                      11709 non-null  object
11  zip_code                     11709 non-null  int64
12  area_code                    11709 non-null  int64
13  lat                          11709 non-null  float64
14  lng                          11709 non-null  float64
15  ALand                        11709 non-null  int64
16  AWater                       11709 non-null  int64
17  pop                          11709 non-null  int64
18  male_pop                     11709 non-null  int64
19  female_pop                   11709 non-null  int64
20  rent_mean                    11561 non-null  float64
21  rent_median                  11561 non-null  float64
22  rent_stdev                   11561 non-null  float64
23  rent_sample_weight           11561 non-null  float64
24  rent_samples                 11561 non-null  float64
25  rent_gt_10                   11560 non-null  float64
26  rent_gt_15                   11560 non-null  float64
27  rent_gt_20                   11560 non-null  float64
28  rent_gt_25                   11560 non-null  float64
29  rent_gt_30                   11560 non-null  float64
30  rent_gt_35                   11560 non-null  float64
31  rent_gt_40                   11560 non-null  float64
32  rent_gt_50                   11560 non-null  float64
33  universe_samples             11709 non-null  int64
34  used_samples                 11709 non-null  int64
35  hi_mean                      11587 non-null  float64
36  hi_median                    11587 non-null  float64
37  hi_stdev                     11587 non-null  float64
38  hi_sample_weight             11587 non-null  float64
39  hi_samples                   11587 non-null  float64
40  family_mean                  11573 non-null  float64
41  family_median                11573 non-null  float64
42  family_stdev                 11573 non-null  float64
43  family_sample_weight         11573 non-null  float64
44  family_samples               11573 non-null  float64
45  hc_mortgage_mean             11441 non-null  float64
46  hc_mortgage_median           11441 non-null  float64
47  hc_mortgage_stdev            11441 non-null  float64
48  hc_mortgage_sample_weight    11441 non-null  float64
49  hc_mortgage_samples          11441 non-null  float64
50  hc_mean                      11419 non-null  float64
51  hc_median                    11419 non-null  float64
52  hc_stdev                     11419 non-null  float64
53  hc_samples                   11419 non-null  float64
```

```
 54  hc_sample_weight             11419 non-null  float64
 55  home_equity_second_mortgage  11489 non-null  float64
 56  second_mortgage              11489 non-null  float64
 57  home_equity                  11489 non-null  float64
 58  debt                         11489 non-null  float64
 59  second_mortgage_cdf          11489 non-null  float64
 60  home_equity_cdf              11489 non-null  float64
 61  debt_cdf                     11489 non-null  float64
 62  hs_degree                    11624 non-null  float64
 63  hs_degree_male               11620 non-null  float64
 64  hs_degree_female             11604 non-null  float64
 65  male_age_mean                11625 non-null  float64
 66  male_age_median              11625 non-null  float64
 67  male_age_stdev               11625 non-null  float64
 68  male_age_sample_weight       11625 non-null  float64
 69  male_age_samples             11625 non-null  float64
 70  female_age_mean              11613 non-null  float64
 71  female_age_median            11613 non-null  float64
 72  female_age_stdev             11613 non-null  float64
 73  female_age_sample_weight     11613 non-null  float64
 74  female_age_samples           11613 non-null  float64
 75  pct_own                      11587 non-null  float64
 76  married                      11625 non-null  float64
 77  married_snp                  11625 non-null  float64
 78  separated                    11625 non-null  float64
 79  divorced                     11625 non-null  float64
dtypes: float64(61), int64(13), object(6)
memory usage: 7.1+ MB
```

[100]:

```
##Figure out the primary key and look for the requirement of indexing.
#make UID as index
```

[101]:

```
train.set_index(keys=['UID'],inplace=True)
```

[102]:

```
test.set_index(keys=['UID'],inplace=True)
```

[103]:

```
test.head()
```

[103]:

```
        BLOCKID  SUMLEVEL  COUNTYID  STATEID         state state_ab  \
UID
255504      NaN       140       163       26      Michigan       MI
252676      NaN       140         1       23         Maine       ME
276314      NaN       140        15       42  Pennsylvania       PA
248614      NaN       140       231       21      Kentucky       KY
286865      NaN       140       355       48         Texas       TX

                    city                  place     type primary  …  \
```

```
UID                                                                      …
255504         Detroit  Dearborn Heights City      CDP   tract  …
252676          Auburn            Auburn City     City   tract  …
276314       Pine City              Millerton  Borough   tract  …
248614      Monticello        Monticello City     City   tract  …
286865  Corpus Christi                  Edroy     Town   tract  …

        female_age_mean  female_age_median  female_age_stdev  \
UID
255504         34.78682           33.75000          21.58531
252676         44.23451           46.66667          22.37036
276314         41.62426           44.50000          22.86213
248614         44.81200           48.00000          21.03155
286865         40.66618           42.66667          21.30900

        female_age_sample_weight  female_age_samples  pct_own  married  \
UID
255504                 416.48097              1938.0  0.70252  0.28217
252676                 532.03505              1950.0  0.85128  0.64221
276314                 453.11959              1879.0  0.81897  0.59961
248614                 263.94320              1081.0  0.84609  0.56953
286865                 709.90829              2956.0  0.79077  0.57620

        married_snp  separated  divorced
UID
255504      0.05910    0.03813   0.14299
252676      0.02338    0.00000   0.13377
276314      0.01746    0.01358   0.10026
248614      0.05492    0.04694   0.12489
286865      0.01726    0.00588   0.16379

[5 rows x 79 columns]
```

[104]:
```
##Gauge the fill rate of variables and devise plan for missing value treatemnt.
↪please explain explicitly the reason for the treatment.chosen for varibales
```

[105]:
```
train.isna().sum().any()
```

[105]:
```
True
```

[106]:
```
test.isna().sum().any()
```

[106]:
```
True
```

[107]:
```
#print the only value which we have missing values
train.isna().sum()[test.isna().sum()>0]
```

[107]:
```
BLOCKID                        27321
rent_mean                        314
rent_median                      314
rent_stdev                       314
rent_sample_weight               314
rent_samples                     314
rent_gt_10                       314
rent_gt_15                       314
rent_gt_20                       314
rent_gt_25                       314
rent_gt_30                       314
rent_gt_35                       314
rent_gt_40                       314
rent_gt_50                       314
hi_mean                          268
hi_median                        268
hi_stdev                         268
hi_sample_weight                 268
hi_samples                       268
family_mean                      298
family_median                    298
family_stdev                     298
family_sample_weight             298
family_samples                   298
hc_mortgage_mean                 573
hc_mortgage_median               573
hc_mortgage_stdev                573
hc_mortgage_sample_weight        573
hc_mortgage_samples              573
hc_mean                          600
hc_median                        600
hc_stdev                         600
hc_samples                       600
hc_sample_weight                 600
home_equity_second_mortgage      457
second_mortgage                  457
home_equity                      457
debt                             457
second_mortgage_cdf              457
home_equity_cdf                  457
debt_cdf                         457
hs_degree                        190
hs_degree_male                   200
hs_degree_female                 223
male_age_mean                    189
male_age_median                  189
male_age_stdev                   189
```

```
male_age_sample_weight          189
male_age_samples                189
female_age_mean                 206
female_age_median               206
female_age_stdev                206
female_age_sample_weight        206
female_age_samples              206
pct_own                         268
married                         191
married_snp                     191
separated                       191
divorced                        191
dtype: int64
```

[108]:
```
train.isna().sum()[test.isna().sum()>0].shape
```

[108]:
```
(59,)
```

[109]:
```
test.isna().sum()[test.isna().sum()>0].shape
```

[109]:
```
(59,)
```

[110]:
```
#calculate precentage for missing values
precentage_train=train.isna().sum()/len(train)*100
```

[111]:
```
precentage_train
```

[111]:
```
BLOCKID        100.000000
SUMLEVEL         0.000000
COUNTYID         0.000000
STATEID          0.000000
state            0.000000
                   …
pct_own          0.980930
married          0.699096
married_snp      0.699096
separated        0.699096
divorced         0.699096
Length: 79, dtype: float64
```

[112]:
```
precentage_train=pd.DataFrame(precentage_train,columns=['precentage og missing␣
 ↪value'])
```

[113]:
```
precentage_train
```

[113]:
```
             precentage og missing value
BLOCKID                       100.000000
```

```
SUMLEVEL                         0.000000
COUNTYID                         0.000000
STATEID                          0.000000
state                            0.000000
…                                   …
pct_own                          0.980930
married                          0.699096
married_snp                      0.699096
separated                        0.699096
divorced                         0.699096

[79 rows x 1 columns]
```

[114]:
```
precentage_train.sort_values(by=['precentage og missing␣
 ↪value'],inplace=True,ascending=False)
```

[115]:
```
precentage_train
```

[115]:
```
             precentage og missing value
BLOCKID                       100.000000
hc_samples                      2.196113
hc_mean                         2.196113
hc_median                       2.196113
hc_stdev                        2.196113
…                                    …
state                           0.000000
zip_code                        0.000000
city                            0.000000
place                           0.000000
state_ab                        0.000000

[79 rows x 1 columns]
```

[116]:
```
precentage_test=test.isna().sum()/len(test)*100
```

[117]:
```
precentage_test=pd.DataFrame(precentage_test,columns=['precentage og missing␣
 ↪value'])
```

[118]:
```
precentage_test.sort_values(by=['precentage og missing␣
 ↪value'],inplace=True,ascending=False)
```

[119]:
```
precentage_test
```

[119]:
```
             precentage og missing value
BLOCKID                       100.000000
hc_samples                      2.476727
hc_mean                         2.476727
```

```
hc_median                       2.476727
hc_stdev                        2.476727
…                                  …
type                            0.000000
place                           0.000000
city                            0.000000
state                           0.000000
state_ab                        0.000000

[79 rows x 1 columns]
```

[120]:
```
#Dropping block id and sumlevel
train.drop(columns=['BLOCKID','SUMLEVEL'],inplace=True)
```

[121]:
```
test.drop(columns=['BLOCKID','SUMLEVEL'],inplace=True)
```

[122]:
```
#colums in train data which are missing values
missing_value_train=[]
for col in train.columns:
    if train[col].isna().sum()!=0:
        missing_value_train.append(col)
```

[123]:
```
missing_values_test=[]
for col in test.columns:
    if test[col].isna().sum()!=0:
        missing_values_test.append(col)
```

[124]:
```
for col in train.columns:
    if col in (missing_value_train):
        train[col].replace(np.nan,train[col].mean(),inplace=True)
```

[125]:
```
for col in test.columns:
    if col in (missing_values_test):
        test[col].replace(np.nan,test[col].mean(),inplace=True)
```

[126]:
```
train.isna().sum().any()
```

[126]: False

[127]:
```
test.isna().sum().any()
```

[127]: False

[128]:
```
##Exploratory Data Analysis (EDA)
#Explore the top 2,500 locations where the percentage of households with a␣
↪second mortgage is the highest and percent ownership is above 10 percent.
```

```
#Visualize using geo-map. You may keep the upper limit for the percent of␣
↪households with a second mortgage to 50 percent
```

[129]:
```
pip install pandasql
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pandasql in
/home/labsuser/.local/lib/python3.7/site-packages (0.7.3)
Requirement already satisfied: sqlalchemy in /usr/local/lib/python3.7/site-
packages (from pandasql) (1.3.15)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/site-packages
(from pandasql) (1.21.5)
Requirement already satisfied: pandas in /usr/local/lib/python3.7/site-packages
(from pandasql) (1.1.5)
Requirement already satisfied: python-dateutil>=2.7.3 in
/usr/local/lib/python3.7/site-packages (from pandas->pandasql) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.7/site-
packages (from pandas->pandasql) (2019.3)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/site-
packages (from python-dateutil>=2.7.3->pandas->pandasql) (1.14.0)
WARNING: You are using pip version 22.0.3; however, version 23.1.2 is
available.
You should consider upgrading via the '/usr/local/bin/python3.7 -m pip install
--upgrade pip' command.
Note: you may need to restart the kernel to use updated packages.
```

[130]:
```
train.columns
```

[130]:
```
Index(['COUNTYID', 'STATEID', 'state', 'state_ab', 'city', 'place', 'type',
       'primary', 'zip_code', 'area_code', 'lat', 'lng', 'ALand', 'AWater',
       'pop', 'male_pop', 'female_pop', 'rent_mean', 'rent_median',
       'rent_stdev', 'rent_sample_weight', 'rent_samples', 'rent_gt_10',
       'rent_gt_15', 'rent_gt_20', 'rent_gt_25', 'rent_gt_30', 'rent_gt_35',
       'rent_gt_40', 'rent_gt_50', 'universe_samples', 'used_samples',
       'hi_mean', 'hi_median', 'hi_stdev', 'hi_sample_weight', 'hi_samples',
       'family_mean', 'family_median', 'family_stdev', 'family_sample_weight',
       'family_samples', 'hc_mortgage_mean', 'hc_mortgage_median',
       'hc_mortgage_stdev', 'hc_mortgage_sample_weight', 'hc_mortgage_samples',
       'hc_mean', 'hc_median', 'hc_stdev', 'hc_samples', 'hc_sample_weight',
       'home_equity_second_mortgage', 'second_mortgage', 'home_equity', 'debt',
       'second_mortgage_cdf', 'home_equity_cdf', 'debt_cdf', 'hs_degree',
       'hs_degree_male', 'hs_degree_female', 'male_age_mean',
       'male_age_median', 'male_age_stdev', 'male_age_sample_weight',
       'male_age_samples', 'female_age_mean', 'female_age_median',
       'female_age_stdev', 'female_age_sample_weight', 'female_age_samples',
```

```
       'pct_own', 'married', 'married_snp', 'separated', 'divorced'],
      dtype='object')
```

[131]:
```
from pandasql import sqldf
q1="Select place,pct_own,second_mortgage,lat,lng from train where pct_own>0.10␣
↪and second_mortgage<0.5 order by second_mortgage DESC LIMIT 2500;"
```

[132]:
```
Query_fun=lambda q:sqldf(q,globals()) # SHIFT+TAB
df_train_location=Query_fun(q1)
```

[133]:
```
df_train_location
```

[133]:
```
                  place  pct_own  second_mortgage        lat         lng
0        Worcester City  0.20247          0.43363  42.254262  -71.800347
1          Harbor Hills  0.15618          0.31818  40.751809  -73.853582
2           Glen Burnie  0.22380          0.30212  39.127273  -76.635265
3        Egypt Lake-leto  0.11618          0.28972  28.029063  -82.495395
4            Lincolnwood  0.14228          0.28899  41.967289  -87.652434
…                    …        …               …          …           …
2495     Marina Del Rey  0.44682          0.06818  33.983203 -118.466139
2496       Raleigh City  0.12827          0.06818  35.757135  -78.704288
2497           Lochearn  0.84707          0.06815  39.353095  -76.733315
2498       Manteca City  0.67116          0.06814  37.732143 -121.242902
2499  Philadelphia City  0.70507          0.06814  40.039070  -75.125135

[2500 rows x 5 columns]
```

[134]:
```
train['bad_debt'] = train['second_mortgage'] + train['home_equity'] -␣
↪train['home_equity_second_mortgage']
```

[135]:
```
#Create Box and whisker plot and analyze the distribution for 2nd mortgage,␣
↪home equity, good debt, and bad debt for different cities
```

[136]:
```
df_ham=train.loc[train['city']=='Hamilton']
df_Man=train.loc[train['city']=='Manhattan']
```

[137]:
```
df_box_city=pd.concat([df_ham,df_Man])
```

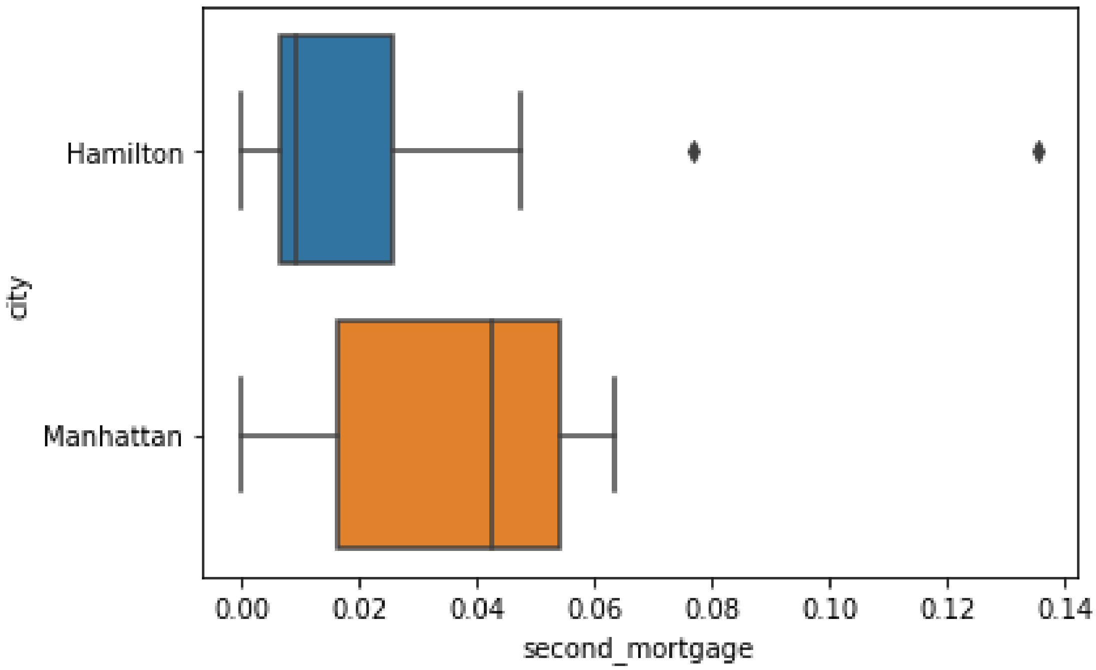
[138]:
```
df_box_city.tail()
```

[138]:
```
        COUNTYID  STATEID     state state_ab       city           place  \
UID
247218       161       20    Kansas       KS  Manhattan  Manhattan City
247221       161       20    Kansas       KS  Manhattan  Manhattan City
247222       161       20    Kansas       KS  Manhattan  Manhattan City
247226       161       20    Kansas       KS  Manhattan  Manhattan City
245107       197       17  Illinois       IL  Manhattan       Manhattan
```

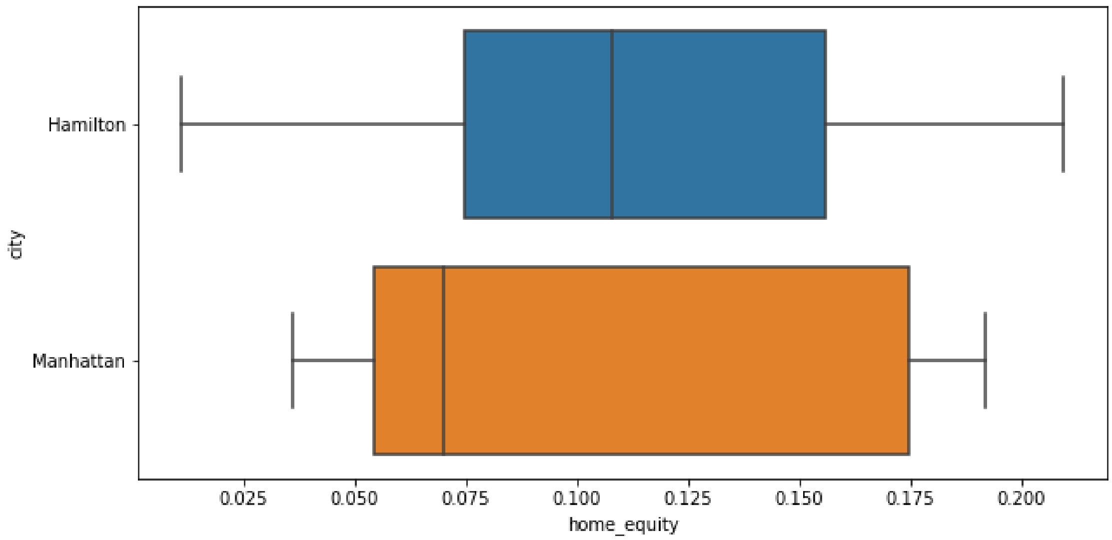
```
           type primary  zip_code  area_code  …  female_age_median  \
UID                                           …
247218     City   tract     66502        785  …           21.58333
247221     City   tract     66502        785  …           22.58333
247222     City   tract     66502        785  …           28.50000
247226     City   tract     66503        785  …           33.75000
245107  Village   tract     60442        815  …           33.41667

        female_age_stdev  female_age_sample_weight  female_age_samples  \
UID
247218          11.10484                1854.48652              3051.0
247221          13.58297                 881.65612              1949.0
247222          22.97004                 401.88911              1171.0
247226          21.63916                 476.01198              1740.0
245107          20.40910                 579.66259              2491.0

        pct_own  married  married_snp  separated  divorced  bad_debt
UID
247218  0.13046  0.12356      0.00000    0.00000   0.03109   0.05426
247221  0.16457  0.13823      0.02133    0.01231   0.08080   0.06985
247222  0.33214  0.29648      0.03015    0.00000   0.11357   0.03581
247226  0.60948  0.61940      0.02572    0.00220   0.01837   0.19200
245107  0.93544  0.67987      0.00597    0.00597   0.04775   0.17486

[5 rows x 78 columns]
```

[139]:

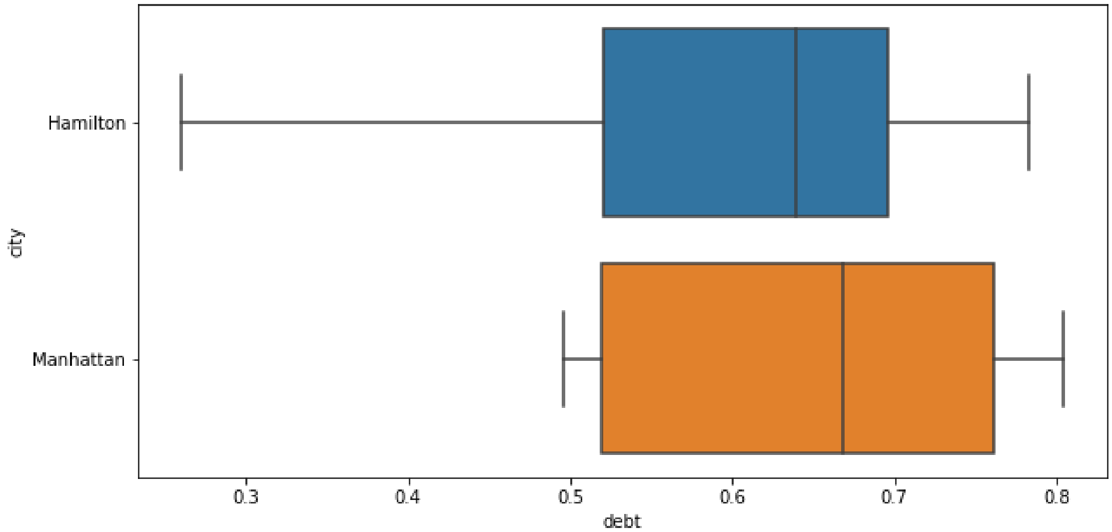
```
# create a boxplot city & second mortgage
sns.boxplot(data=df_box_city,x='second_mortgage',y='city')
plt.show()
```



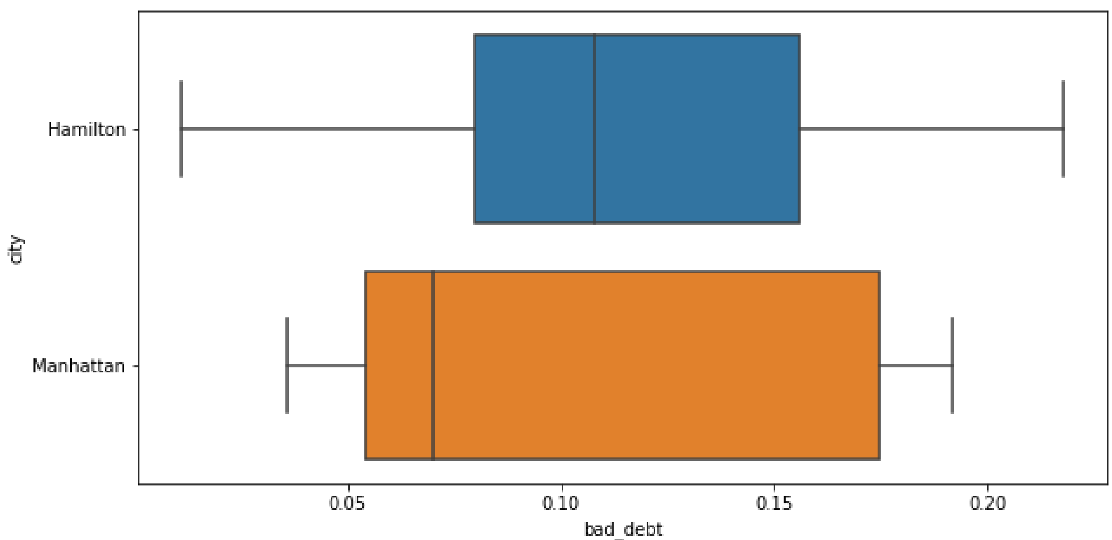
[140]:
```
#city vs home equity
plt.figure(figsize=(10,5))
sns.boxplot(data=df_box_city,x='home_equity', y='city')
plt.show()
```



[141]:
```
#debt vs city
plt.figure(figsize=(10,5))
sns.boxplot(data=df_box_city,x='debt', y='city')
plt.show()
```



[142]:
```
#bad debt vs city
plt.figure(figsize=(10,5))
sns.boxplot(data=df_box_city,x='bad_debt', y='city')
plt.show()
```



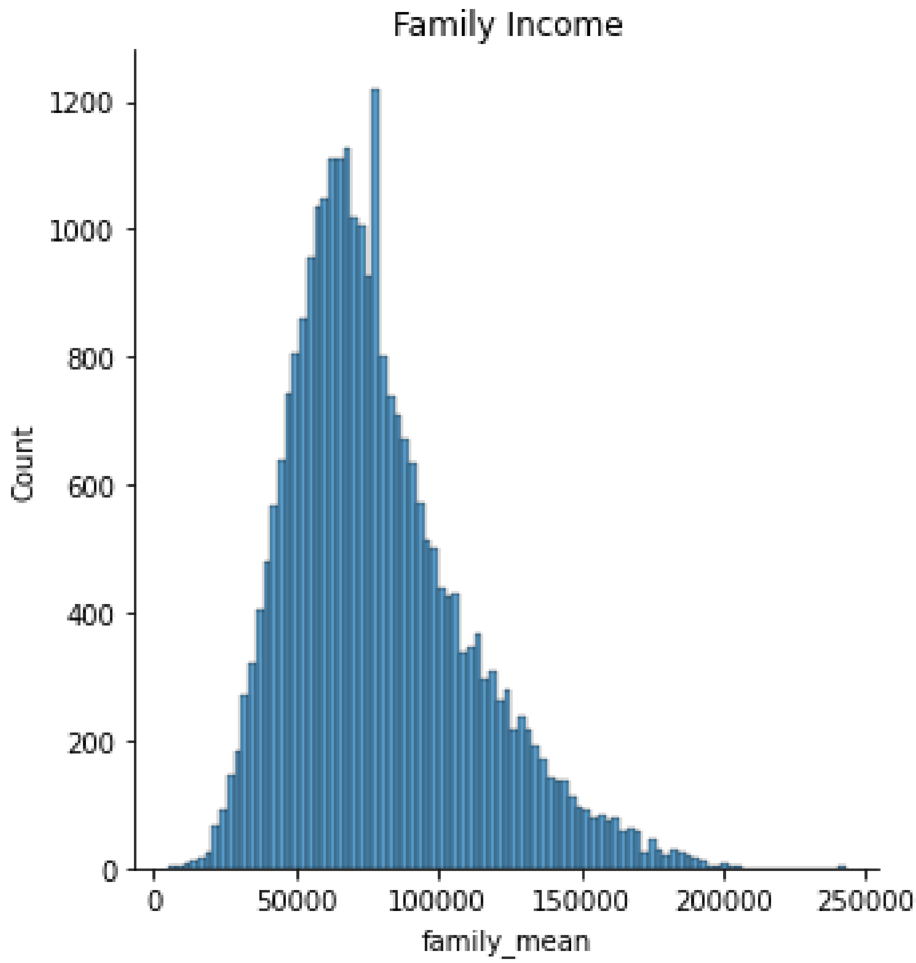
[143]:
```
#Create a collated income distribution chart for family income, house hold␣
↪income, and remaining income
```

[144]:
```
train.columns
```

[144]:
```
Index(['COUNTYID', 'STATEID', 'state', 'state_ab', 'city', 'place', 'type',
       'primary', 'zip_code', 'area_code', 'lat', 'lng', 'ALand', 'AWater',
       'pop', 'male_pop', 'female_pop', 'rent_mean', 'rent_median',
       'rent_stdev', 'rent_sample_weight', 'rent_samples', 'rent_gt_10',
       'rent_gt_15', 'rent_gt_20', 'rent_gt_25', 'rent_gt_30', 'rent_gt_35',
       'rent_gt_40', 'rent_gt_50', 'universe_samples', 'used_samples',
       'hi_mean', 'hi_median', 'hi_stdev', 'hi_sample_weight', 'hi_samples',
       'family_mean', 'family_median', 'family_stdev', 'family_sample_weight',
       'family_samples', 'hc_mortgage_mean', 'hc_mortgage_median',
       'hc_mortgage_stdev', 'hc_mortgage_sample_weight', 'hc_mortgage_samples',
       'hc_mean', 'hc_median', 'hc_stdev', 'hc_samples', 'hc_sample_weight',
       'home_equity_second_mortgage', 'second_mortgage', 'home_equity', 'debt',
       'second_mortgage_cdf', 'home_equity_cdf', 'debt_cdf', 'hs_degree',
       'hs_degree_male', 'hs_degree_female', 'male_age_mean',
       'male_age_median', 'male_age_stdev', 'male_age_sample_weight',
       'male_age_samples', 'female_age_mean', 'female_age_median',
       'female_age_stdev', 'female_age_sample_weight', 'female_age_samples',
       'pct_own', 'married', 'married_snp', 'separated', 'divorced',
       'bad_debt'],
      dtype='object')
```

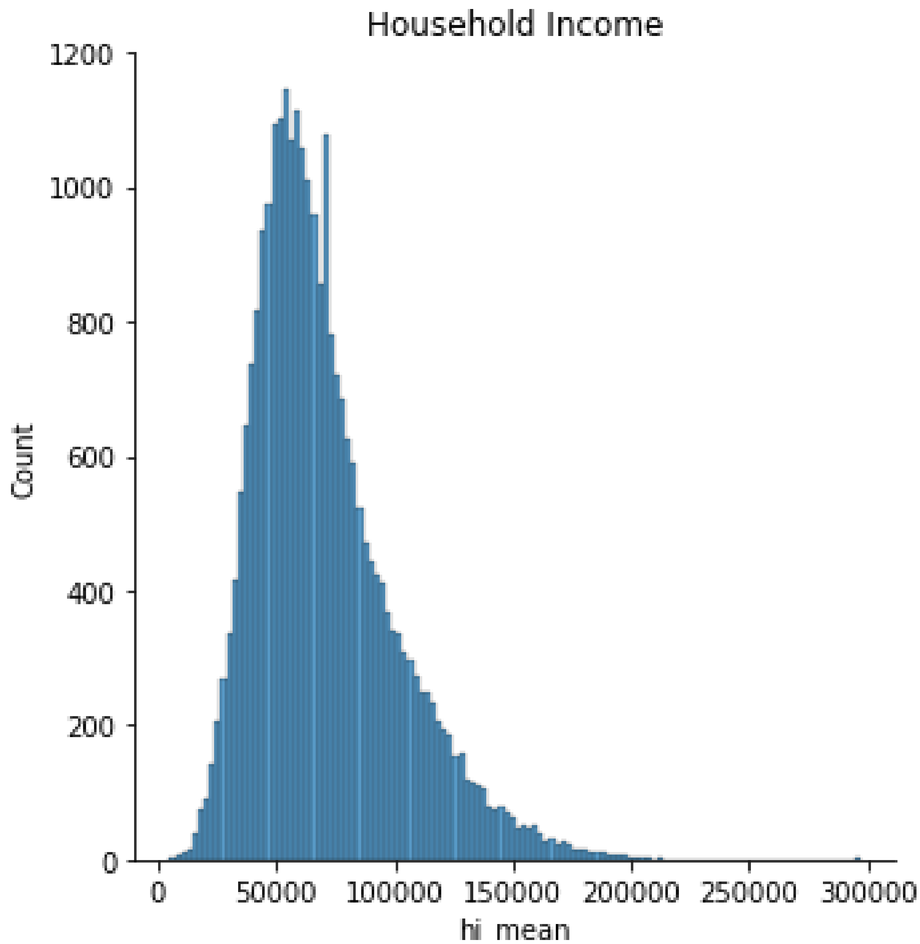
[145]:
```
sns.displot(train['family_mean'])
plt.title('Family Income')
```

[145]:
```
Text(0.5, 1.0, 'Family Income')
```



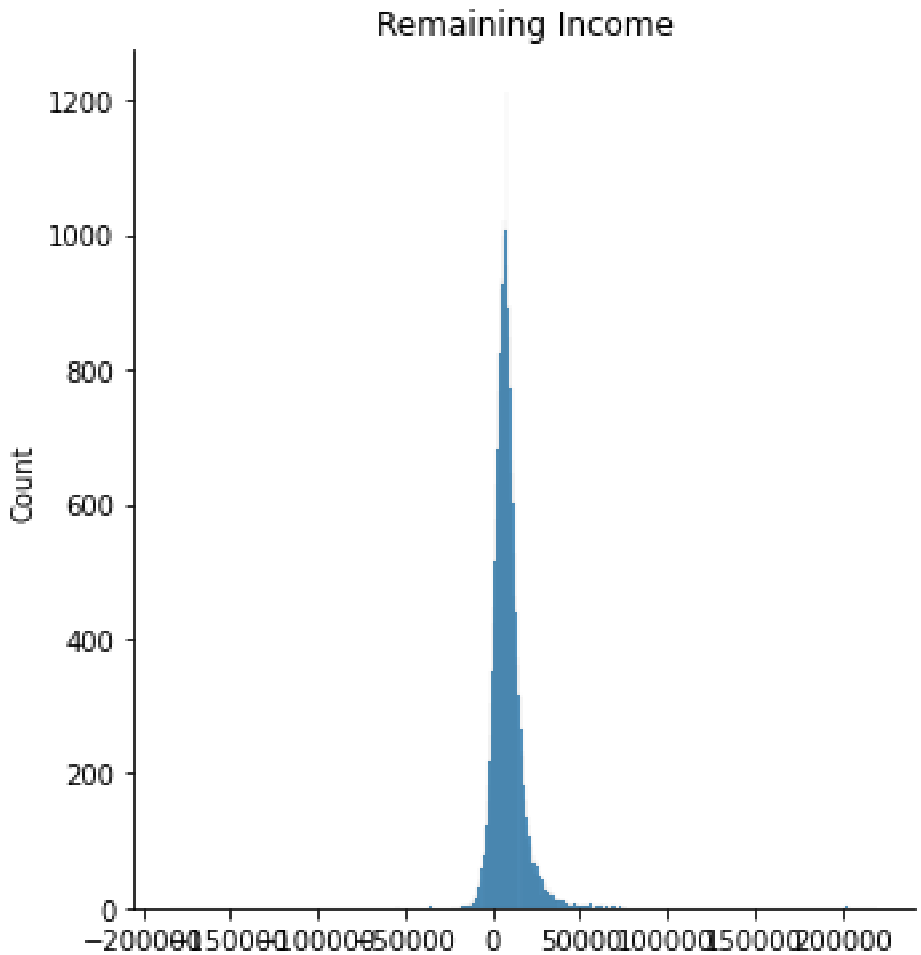
[146]:
```
sns.displot(train['hi_mean'])
plt.title('Household Income')
```

[146]: Text(0.5, 1.0, 'Household Income')



[147]:
```
sns.displot(train['family_mean']-train['hi_mean'])
plt.title('Remaining Income')
```

[147]: Text(0.5, 1.0, 'Remaining Income')



[148]:
```
##Perform EDA and come out with insights into population density and age. You␣
↪may have to derive new fields (make sure to weight averages for accurate␣
↪measurements):

#Use pop and ALand variables to create a new field called population density

#Use male_age_median, female_age_median, male_pop, and female_pop to create a␣
↪new field called median age

#Visualize the findings using appropriate chart type
```

[149]:
```
train.columns
```

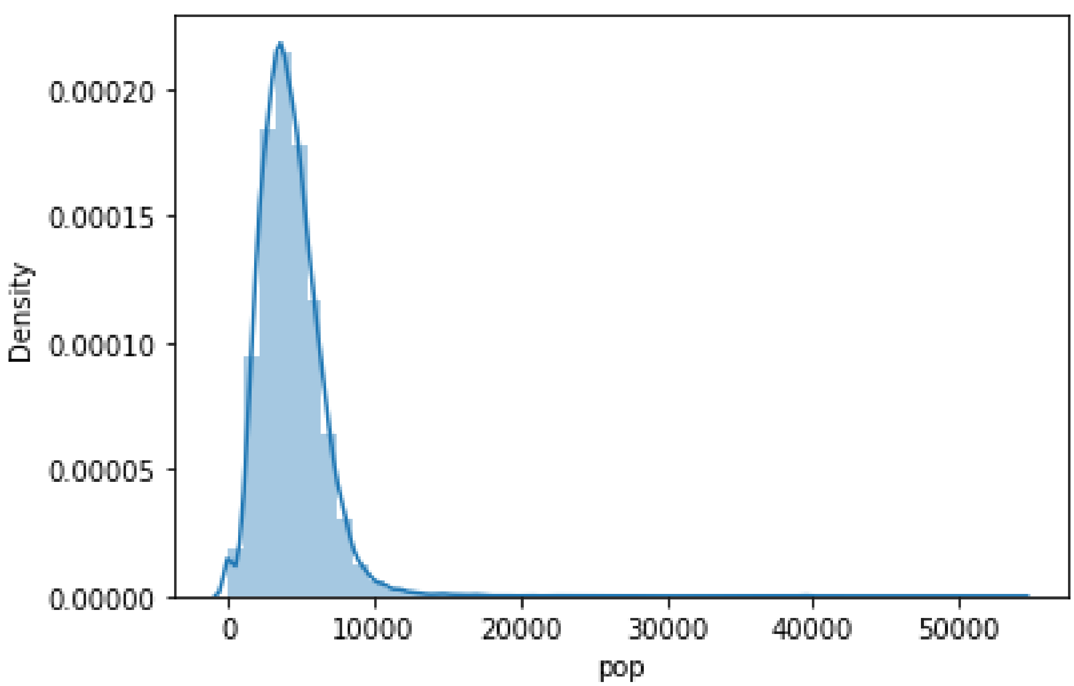
[149]:
```
Index(['COUNTYID', 'STATEID', 'state', 'state_ab', 'city', 'place', 'type',
       'primary', 'zip_code', 'area_code', 'lat', 'lng', 'ALand', 'AWater',
       'pop', 'male_pop', 'female_pop', 'rent_mean', 'rent_median',
       'rent_stdev', 'rent_sample_weight', 'rent_samples', 'rent_gt_10',
```

```
 'rent_gt_15', 'rent_gt_20', 'rent_gt_25', 'rent_gt_30', 'rent_gt_35',
 'rent_gt_40', 'rent_gt_50', 'universe_samples', 'used_samples',
 'hi_mean', 'hi_median', 'hi_stdev', 'hi_sample_weight', 'hi_samples',
 'family_mean', 'family_median', 'family_stdev', 'family_sample_weight',
 'family_samples', 'hc_mortgage_mean', 'hc_mortgage_median',
 'hc_mortgage_stdev', 'hc_mortgage_sample_weight', 'hc_mortgage_samples',
 'hc_mean', 'hc_median', 'hc_stdev', 'hc_samples', 'hc_sample_weight',
 'home_equity_second_mortgage', 'second_mortgage', 'home_equity', 'debt',
 'second_mortgage_cdf', 'home_equity_cdf', 'debt_cdf', 'hs_degree',
 'hs_degree_male', 'hs_degree_female', 'male_age_mean',
 'male_age_median', 'male_age_stdev', 'male_age_sample_weight',
 'male_age_samples', 'female_age_mean', 'female_age_median',
 'female_age_stdev', 'female_age_sample_weight', 'female_age_samples',
 'pct_own', 'married', 'married_snp', 'separated', 'divorced',
 'bad_debt'],
dtype='object')
```

[150]:
```
sns.distplot(train['pop'])
```

```
/usr/local/lib/python3.7/site-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

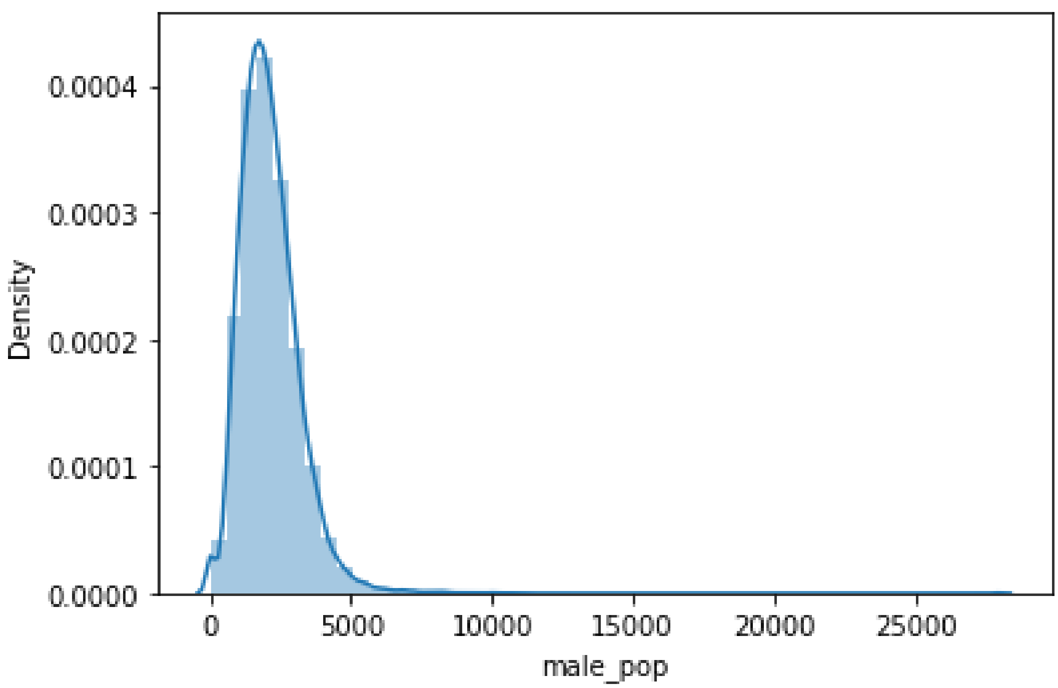
[150]:
```
<AxesSubplot:xlabel='pop', ylabel='Density'>
```



[151]:
```
sns.distplot(train['male_pop'])
```

```
/usr/local/lib/python3.7/site-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

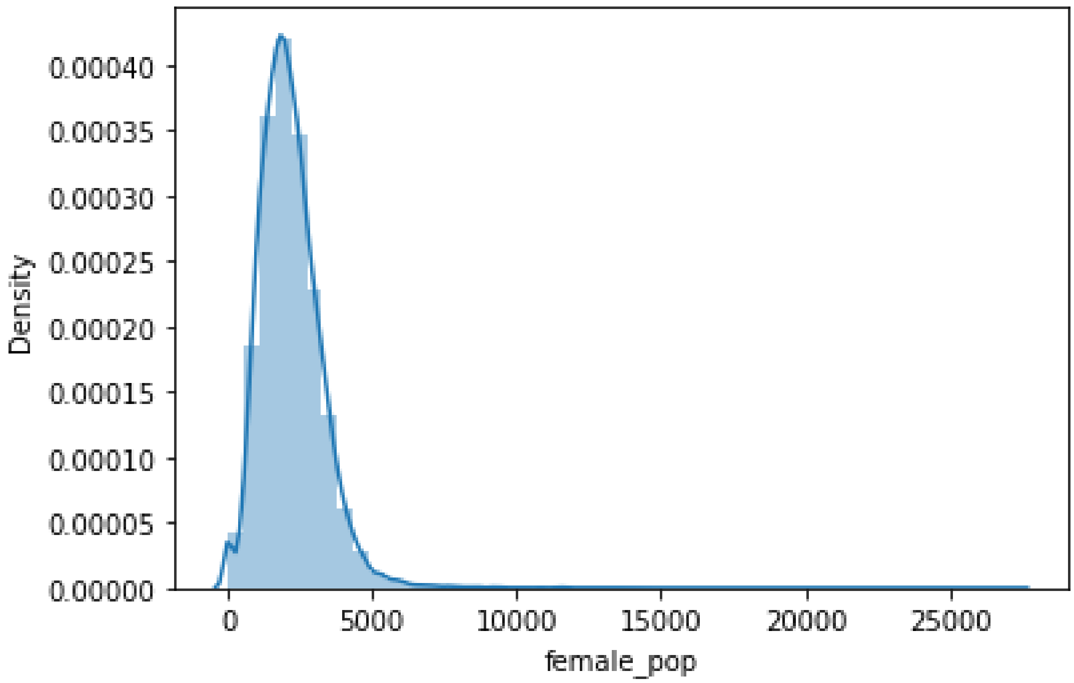
[151]: <AxesSubplot:xlabel='male_pop', ylabel='Density'>



[152]:
```
sns.distplot(train['female_pop'])
```

```
/usr/local/lib/python3.7/site-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

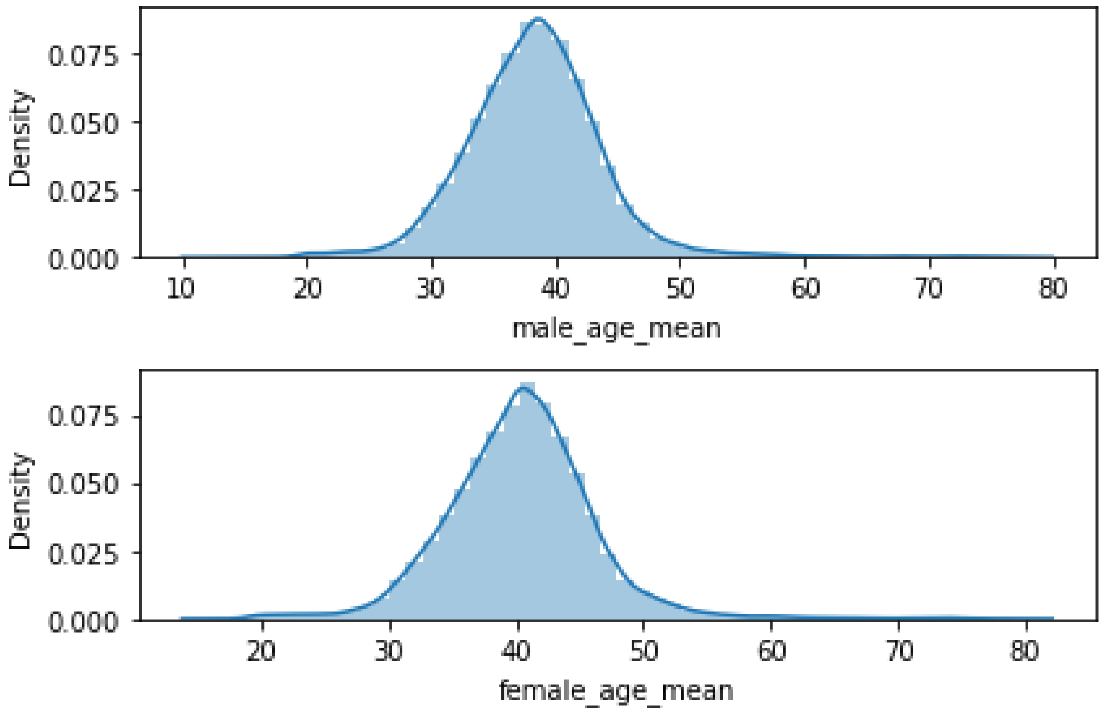
[152]: <AxesSubplot:xlabel='female_pop', ylabel='Density'>



[153]:

```
fig,(ax1,ax2)=plt.subplots(2,1)
plt.subplots_adjust(wspace=0.8,hspace=0.9)
sns.distplot(train['male_age_mean'],ax=ax1)
sns.distplot(train['female_age_mean'],ax=ax2)
plt.tight_layout()
plt.show()
```

```
/usr/local/lib/python3.7/site-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/site-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```



[154]:
```
train['pop_density']=train['pop']/train['ALand']
```

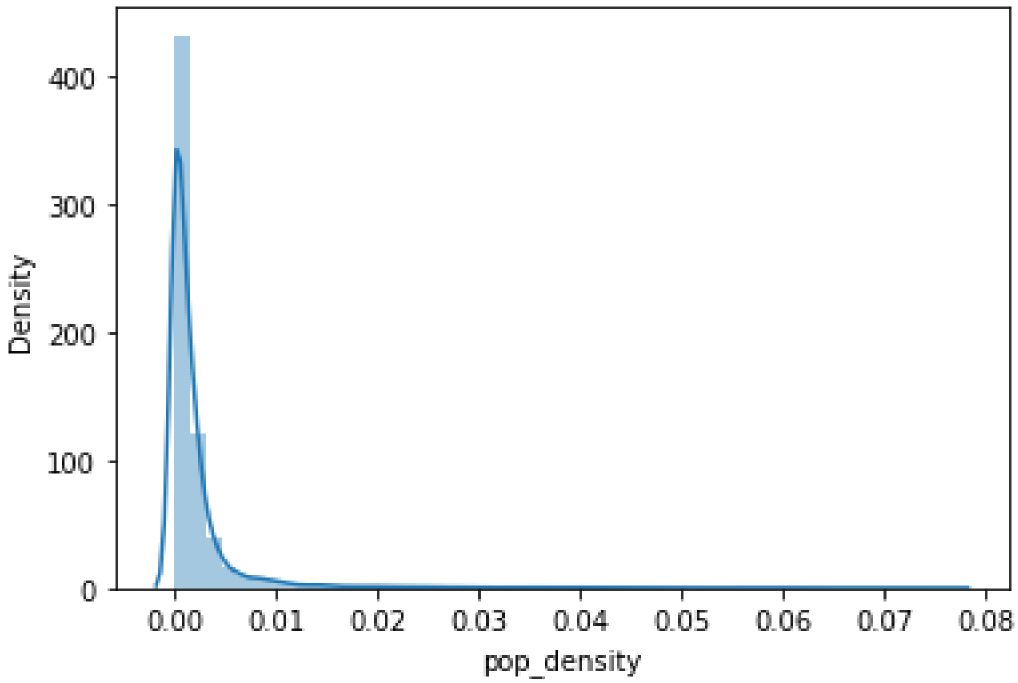
[155]:
```
test['pop_density']=test['pop']/test['ALand']
```

[156]:
```
train['pop_density']
```

[156]:
```
UID
267822    0.000026
246444    0.001687
245683    0.000099
279653    0.002442
247218    0.002207
            …
279212    0.002650
277856    0.000818
233000    0.000002
287425    0.000619
265371    0.000478
Name: pop_density, Length: 27321, dtype: float64
```

[157]:
```
# check population density
sns.distplot(train['pop_density'])
plt.show()
```

```
/usr/local/lib/python3.7/site-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

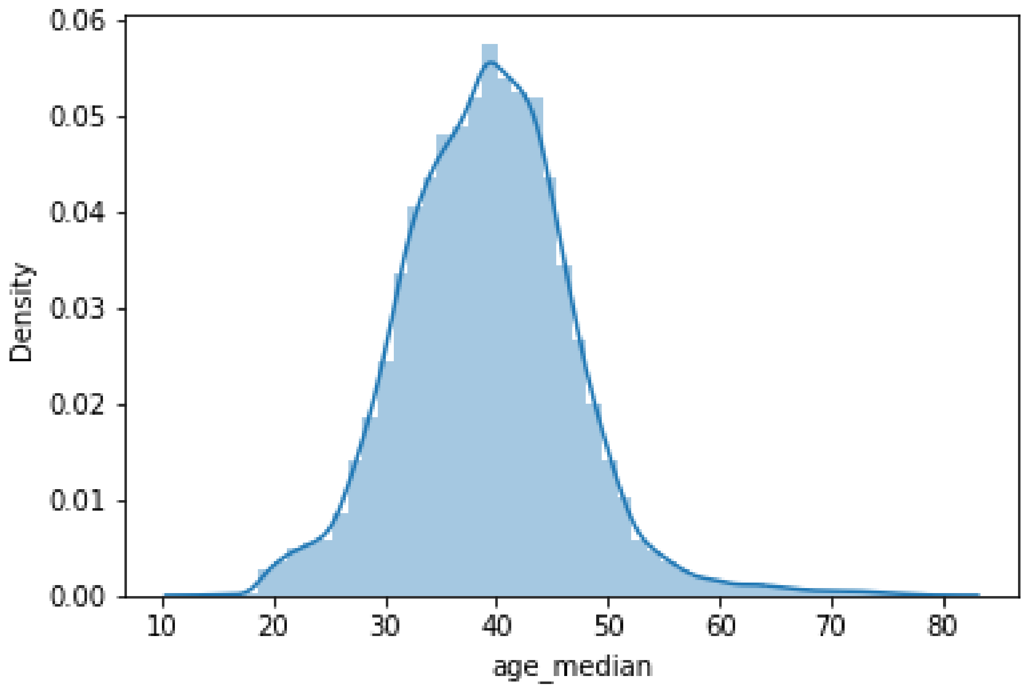


[158]:
```
train['age_median']=(train['male_age_median']+train['female_age_median'])/2
```

[159]:
```
test['age_median']=(test['male_age_median']+test['female_age_median'])/2
```

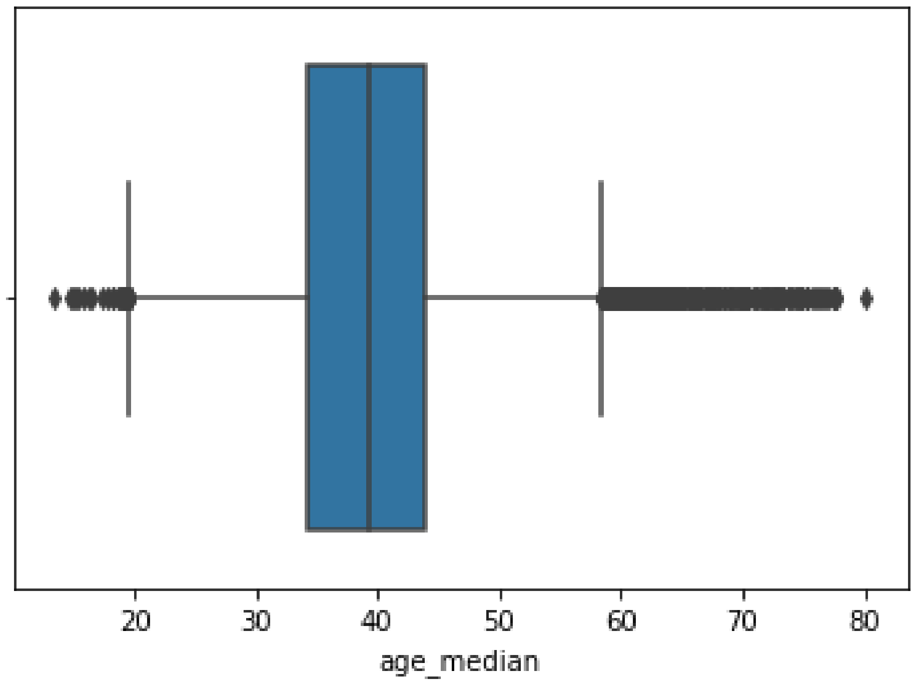
[160]:
```
sns.distplot(train['age_median'])
plt.show()
```

```
/usr/local/lib/python3.7/site-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```



[161]:
```
sns.boxplot(train['age_median'])
plt.show()
```

```
/usr/local/lib/python3.7/site-packages/seaborn/_decorators.py:43: FutureWarning:
Pass the following variable as a keyword arg: x. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an
explicit keyword will result in an error or misinterpretation.
  FutureWarning
```




```
[162]: #Create bins for population into a new variable by selecting appropriate class␣
       ↪interval so that the number of categories don't exceed 5 for the ease of␣
       ↪analysis.

       #Analyze the married, separated, and divorced population for these population␣
       ↪brackets

       #Visualize using appropriate chart type
```

```
[163]: train['pop_bins']=pd.cut(train['pop'],bins=5,labels=['very␣
       ↪low','low','medium','high','very high'])
```

```
[164]: train['pop_bins'].value_counts()
```

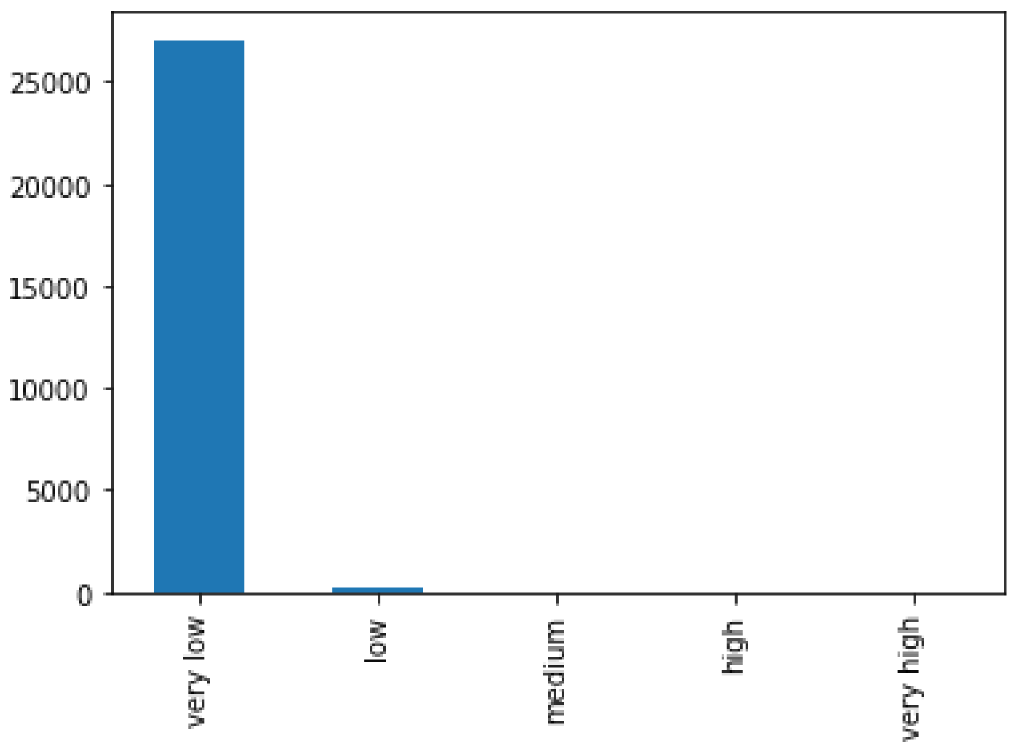
```
[164]: very low     27058
       low            246
       medium           9
       high             7
       very high        1
       Name: pop_bins, dtype: int64
```

```
[165]: train['pop'].describe()
```

[165]:
```
count    27321.000000
mean      4316.032685
std       2169.226173
min          0.000000
25%       2885.000000
50%       4042.000000
75%       5430.000000
max      53812.000000
Name: pop, dtype: float64
```

[166]:
```
train['pop_bins'].value_counts().plot(kind='bar')
```

[166]: <AxesSubplot:>



[167]:
```
train.groupby(by='pop_bins')[['married','separated', 'divorced']].count()
```

[167]:

pop_bins	married	separated	divorced
very low	27058	27058	27058
low	246	246	246
medium	9	9	9
high	7	7	7
very high	1	1	1

[168]:
```
train.groupby(by='pop_bins')[['married','separated', 'divorced']].
↪agg(['sum','mean','median','count'])
```

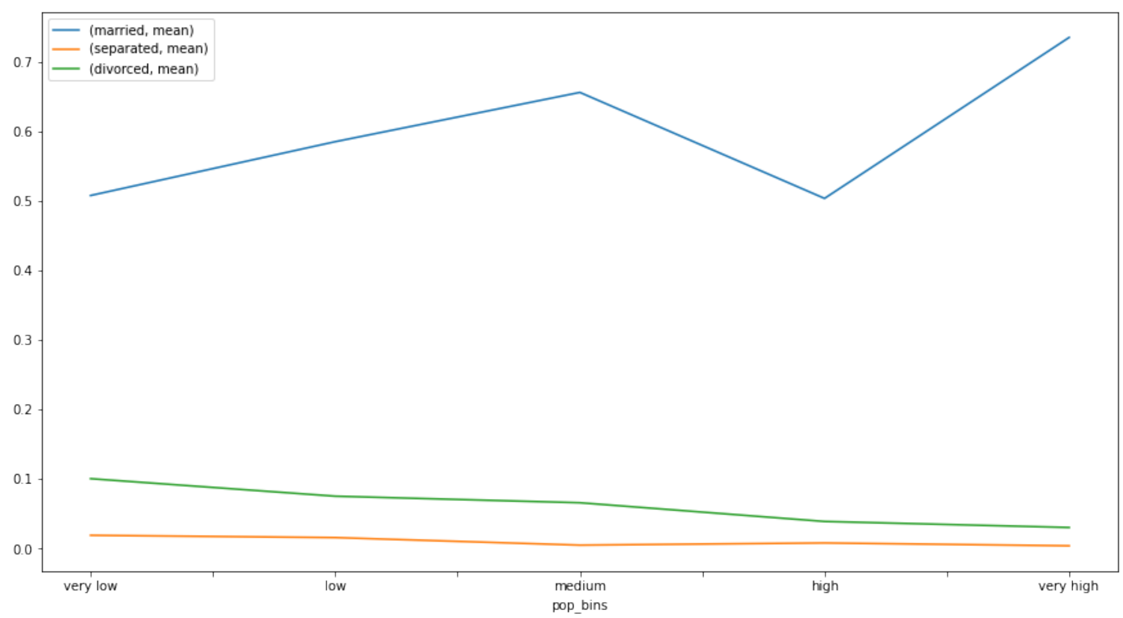
[168]:

	married				separated	
	sum	mean	median	count	sum	mean
pop_bins						
very low	13733.22489	0.507548	0.524680	27058	517.52126	0.019126
low	143.88385	0.584894	0.593135	246	3.89480	0.015833
medium	5.90163	0.655737	0.618710	9	0.04503	0.005003
high	3.52351	0.503359	0.335660	7	0.05699	0.008141
very high	0.73474	0.734740	0.734740	1	0.00405	0.004050

			divorced			
	median	count	sum	mean	median	count
pop_bins						
very low	0.013650	27058	2719.430721	0.100504	0.096020	27058
low	0.011195	246	18.535600	0.075348	0.070045	246
medium	0.004120	9	0.593340	0.065927	0.064890	9
high	0.002500	7	0.273210	0.039030	0.010320	7
very high	0.004050	1	0.030360	0.030360	0.030360	1

[169]:
```
train.groupby(by='pop_bins')[['married','separated', 'divorced']].agg(['mean']).
↪plot(figsize=(15,8))
plt.legend(loc='best')
```

[169]: <matplotlib.legend.Legend at 0x7f9d9b026f10>



[170]:
```
#Perform correlation analysis for all the relevant variables by creating a␣
↪heatmap. Describe your findings.
```

[171]:
```
rent_state_mean=train.groupby(by='state')['rent_mean'].agg(["mean"])
```

[172]:
```
income_state_mean=train.groupby(by='state')['family_mean'].agg(['mean'])
```

[173]:
```
income_state_mean.head()
```

[173]:
```
                    mean
state
Alabama     67030.064213
Alaska      92136.545109
Arizona     73328.238798
Arkansas    64765.377850
California  87655.470820
```

[174]:
```
# calculate rent percentage
rent_perc=rent_state_mean['mean']/income_state_mean['mean']
```

[175]:
```
#heat map
train.columns
```

[175]:
```
Index(['COUNTYID', 'STATEID', 'state', 'state_ab', 'city', 'place', 'type',
       'primary', 'zip_code', 'area_code', 'lat', 'lng', 'ALand', 'AWater',
       'pop', 'male_pop', 'female_pop', 'rent_mean', 'rent_median',
       'rent_stdev', 'rent_sample_weight', 'rent_samples', 'rent_gt_10',
       'rent_gt_15', 'rent_gt_20', 'rent_gt_25', 'rent_gt_30', 'rent_gt_35',
       'rent_gt_40', 'rent_gt_50', 'universe_samples', 'used_samples',
       'hi_mean', 'hi_median', 'hi_stdev', 'hi_sample_weight', 'hi_samples',
       'family_mean', 'family_median', 'family_stdev', 'family_sample_weight',
       'family_samples', 'hc_mortgage_mean', 'hc_mortgage_median',
       'hc_mortgage_stdev', 'hc_mortgage_sample_weight', 'hc_mortgage_samples',
       'hc_mean', 'hc_median', 'hc_stdev', 'hc_samples', 'hc_sample_weight',
       'home_equity_second_mortgage', 'second_mortgage', 'home_equity', 'debt',
       'second_mortgage_cdf', 'home_equity_cdf', 'debt_cdf', 'hs_degree',
       'hs_degree_male', 'hs_degree_female', 'male_age_mean',
       'male_age_median', 'male_age_stdev', 'male_age_sample_weight',
       'male_age_samples', 'female_age_mean', 'female_age_median',
       'female_age_stdev', 'female_age_sample_weight', 'female_age_samples',
       'pct_own', 'married', 'married_snp', 'separated', 'divorced',
       'bad_debt', 'pop_density', 'age_median', 'pop_bins'],
      dtype='object')
```

[176]:
```
num=train.select_dtypes(exclude='object')
```

[177]:
```
num
```

[177]:

```
        COUNTYID  STATEID  zip_code  area_code        lat         lng  \
UID
267822        53       36     13346        315  42.840812  -75.501524
246444       141       18     46616        574  41.701441  -86.266614
245683        63       18     46122        317  39.792202  -86.515246
279653       127       72       927        787  18.396103  -66.104169
247218       161       20     66502        785  39.195573  -96.569366
…             …       …        …          …         …           …
279212        43       72       769        787  18.076060  -66.358379
277856        91       42     19422        215  40.158138  -75.307271
233000        87        8     80653        970  40.410316 -103.814003
287425       439       48     76034        817  32.904866  -97.162151
265371         3       32     89123        702  36.064754 -115.152237

               ALand    AWater    pop  male_pop  …  female_age_samples  \
UID                                               …
267822  2.021834e+08   1699120   5230      2612  …              2618.0
246444  1.560828e+06    100363   2633      1349  …              1284.0
245683  6.956160e+07    284193   6881      3643  …              3238.0
279653  1.105793e+06         0   2700      1141  …              1559.0
247218  2.554403e+06         0   5637      2586  …              3051.0
…                …        …    …        …  …                 …
279212  6.970300e+05         0   1847       909  …               938.0
277856  5.077337e+06     11786   4155      2116  …              2039.0
233000  1.323262e+09  17577610   2829      1465  …              1364.0
287425  1.865230e+07    158882  11542      5727  …              5815.0
265371  7.796308e+06         0   3726      1815  …              1911.0

        pct_own  married  married_snp  separated  divorced  bad_debt  \
UID
267822  0.79046  0.57851      0.01882    0.01240   0.08770   0.09408
246444  0.52483  0.34886      0.01426    0.01426   0.09030   0.04274
245683  0.85331  0.64745      0.02830    0.01607   0.10657   0.09512
279653  0.65037  0.47257      0.02021    0.02021   0.10106   0.01086
247218  0.13046  0.12356      0.00000    0.00000   0.03109   0.05426
…           …        …           …          …         …         …
279212  0.60422  0.24603      0.03042    0.02249   0.14683   0.00000
277856  0.68072  0.61127      0.05003    0.02473   0.04888   0.20908
233000  0.78508  0.70451      0.01386    0.00520   0.07712   0.07857
287425  0.93970  0.75503      0.02287    0.00915   0.05261   0.14305
265371  0.27912  0.34426      0.03825    0.03005   0.13320   0.18362

        pop_density  age_median  pop_bins
UID
267822     0.000026   44.666665  very low
246444     0.001687   34.791665  very low
245683     0.000099   41.833330  very low
```

```
279653     0.002442   49.750000  very low
247218     0.002207   22.000000  very low
…             …          …        …
279212     0.002650   40.916670  very low
277856     0.000818   39.166665  very low
233000     0.000002   44.166665  very low
287425     0.000619   45.041670       low
265371     0.000478   31.166665  very low

[27321 rows x 75 columns]
```

[178]:
```
num.shape
```

[178]:
```
(27321, 75)
```

[179]:
```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 27321 entries, 267822 to 265371
Data columns (total 81 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   COUNTYID                     27321 non-null  int64
 1   STATEID                      27321 non-null  int64
 2   state                        27321 non-null  object
 3   state_ab                     27321 non-null  object
 4   city                         27321 non-null  object
 5   place                        27321 non-null  object
 6   type                         27321 non-null  object
 7   primary                      27321 non-null  object
 8   zip_code                     27321 non-null  int64
 9   area_code                    27321 non-null  int64
 10  lat                          27321 non-null  float64
 11  lng                          27321 non-null  float64
 12  ALand                        27321 non-null  float64
 13  AWater                       27321 non-null  int64
 14  pop                          27321 non-null  int64
 15  male_pop                     27321 non-null  int64
 16  female_pop                   27321 non-null  int64
 17  rent_mean                    27321 non-null  float64
 18  rent_median                  27321 non-null  float64
 19  rent_stdev                   27321 non-null  float64
 20  rent_sample_weight           27321 non-null  float64
 21  rent_samples                 27321 non-null  float64
 22  rent_gt_10                   27321 non-null  float64
 23  rent_gt_15                   27321 non-null  float64
 24  rent_gt_20                   27321 non-null  float64
```

```
25  rent_gt_25                   27321 non-null  float64
26  rent_gt_30                   27321 non-null  float64
27  rent_gt_35                   27321 non-null  float64
28  rent_gt_40                   27321 non-null  float64
29  rent_gt_50                   27321 non-null  float64
30  universe_samples             27321 non-null  int64
31  used_samples                 27321 non-null  int64
32  hi_mean                      27321 non-null  float64
33  hi_median                    27321 non-null  float64
34  hi_stdev                     27321 non-null  float64
35  hi_sample_weight             27321 non-null  float64
36  hi_samples                   27321 non-null  float64
37  family_mean                  27321 non-null  float64
38  family_median                27321 non-null  float64
39  family_stdev                 27321 non-null  float64
40  family_sample_weight         27321 non-null  float64
41  family_samples               27321 non-null  float64
42  hc_mortgage_mean             27321 non-null  float64
43  hc_mortgage_median           27321 non-null  float64
44  hc_mortgage_stdev            27321 non-null  float64
45  hc_mortgage_sample_weight    27321 non-null  float64
46  hc_mortgage_samples          27321 non-null  float64
47  hc_mean                      27321 non-null  float64
48  hc_median                    27321 non-null  float64
49  hc_stdev                     27321 non-null  float64
50  hc_samples                   27321 non-null  float64
51  hc_sample_weight             27321 non-null  float64
52  home_equity_second_mortgage  27321 non-null  float64
53  second_mortgage              27321 non-null  float64
54  home_equity                  27321 non-null  float64
55  debt                         27321 non-null  float64
56  second_mortgage_cdf          27321 non-null  float64
57  home_equity_cdf              27321 non-null  float64
58  debt_cdf                     27321 non-null  float64
59  hs_degree                    27321 non-null  float64
60  hs_degree_male               27321 non-null  float64
61  hs_degree_female             27321 non-null  float64
62  male_age_mean                27321 non-null  float64
63  male_age_median              27321 non-null  float64
64  male_age_stdev               27321 non-null  float64
65  male_age_sample_weight       27321 non-null  float64
66  male_age_samples             27321 non-null  float64
67  female_age_mean              27321 non-null  float64
68  female_age_median            27321 non-null  float64
69  female_age_stdev             27321 non-null  float64
70  female_age_sample_weight     27321 non-null  float64
71  female_age_samples           27321 non-null  float64
72  pct_own                      27321 non-null  float64
```

```
 73  married                      27321 non-null  float64
 74  married_snp                  27321 non-null  float64
 75  separated                    27321 non-null  float64
 76  divorced                     27321 non-null  float64
 77  bad_debt                     27321 non-null  float64
 78  pop_density                  27321 non-null  float64
 79  age_median                   27321 non-null  float64
 80  pop_bins                     27321 non-null  category
dtypes: category(1), float64(64), int64(10), object(6)
memory usage: 18.2+ MB
```

[180]:

```python
num.corr()
```

[180]:

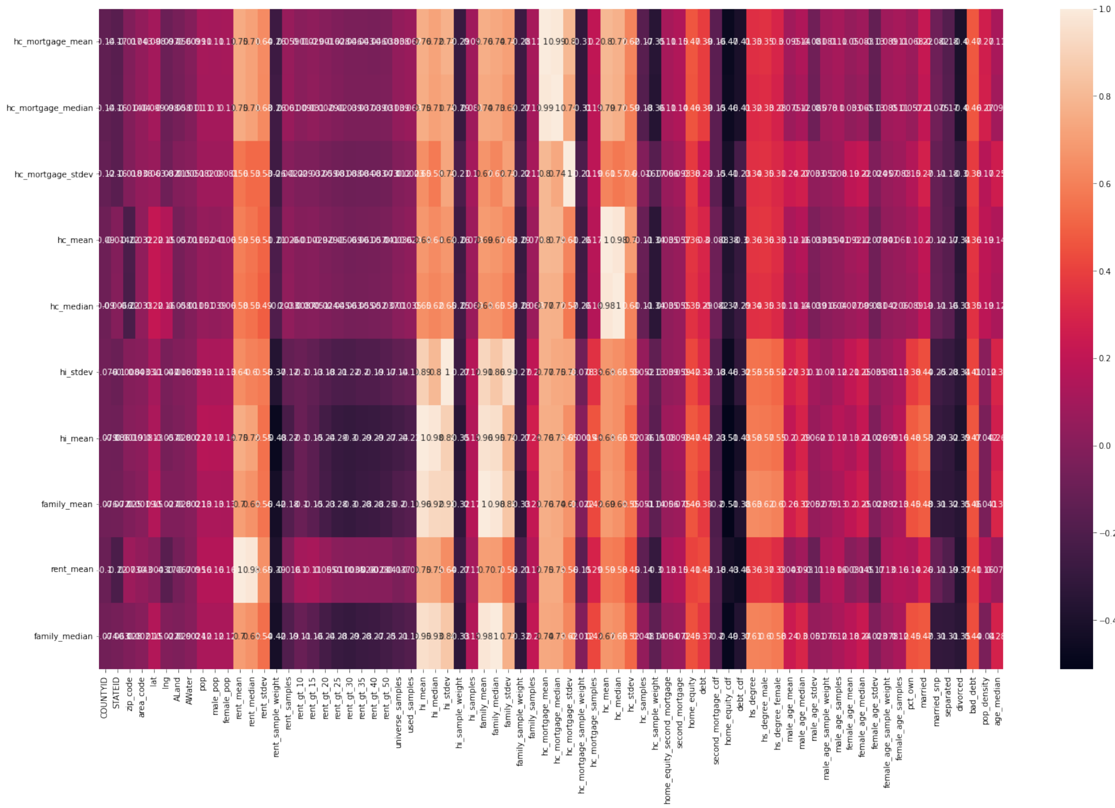
```
             COUNTYID   STATEID  zip_code  area_code       lat       lng  \
COUNTYID     1.000000  0.224549  0.036527   0.067171 -0.149272  0.070414
STATEID      0.224549  1.000000 -0.261465   0.043718  0.109934  0.319964
zip_code     0.036527 -0.261465  1.000000  -0.004681 -0.070775 -0.926708
area_code    0.067171  0.043718 -0.004681   1.000000 -0.125415 -0.013494
lat         -0.149272  0.109934 -0.070775  -0.125415  1.000000  0.025450
…                 …         …         …         …         …         …
separated    0.069059  0.030409 -0.048023   0.022543 -0.138048  0.049228
divorced     0.048850  0.018748  0.043310  -0.043722 -0.056018 -0.004321
bad_debt    -0.125892 -0.151007 -0.069348  -0.003658  0.208792 -0.005876
pop_density -0.080509 -0.013671 -0.119014  -0.030743  0.054513  0.066056
age_median  -0.063521 -0.017172 -0.126150  -0.017118  0.008246  0.104944

                ALand    AWater       pop  male_pop  …  \
COUNTYID     0.015469  0.016550 -0.002662 -0.002615  …
STATEID     -0.017275 -0.026476 -0.036599 -0.040351  …
zip_code     0.072711  0.031679  0.083058  0.099959  …
area_code    0.016563  0.021711  0.031834  0.034387  …
lat          0.100498  0.067660 -0.078283 -0.072763  …
…                 …         …         …         …  …
separated   -0.005904 -0.001208 -0.083182 -0.074929  …
divorced     0.023381  0.007677 -0.160931 -0.146619  …
bad_debt    -0.079618 -0.024112  0.099489  0.092085  …
pop_density -0.044934 -0.013174  0.033740  0.020651  …
age_median   0.042532  0.004878 -0.162499 -0.166810  …

             female_age_sample_weight  female_age_samples   pct_own   married  \
COUNTYID                     0.004587           -0.001227 -0.004632 -0.021428
STATEID                     -0.025104           -0.028238  0.069314  0.025763
zip_code                     0.055497            0.059305 -0.069965  0.030217
area_code                    0.029857            0.031128  0.018877  0.057824
lat                         -0.080855           -0.087667  0.056487  0.035480
…                                 …                 …         …         …
separated                   -0.091913           -0.088709 -0.284877 -0.219686
```

```
divorced                     -0.198491           -0.169450 -0.095413 -0.267833
bad_debt                      0.078159            0.104039  0.134257  0.182985
pop_density                   0.046016            0.040268 -0.426353 -0.248678
age_median                   -0.246096           -0.153775  0.546692  0.495153

             married_snp  separated  divorced  bad_debt  pop_density  \
COUNTYID        0.041710   0.069059  0.048850 -0.125892    -0.080509
STATEID        -0.033283   0.030409  0.018748 -0.151007    -0.013671
zip_code        0.020541  -0.048023  0.043310 -0.069348    -0.119014
area_code       0.022687   0.022543 -0.043722 -0.003658    -0.030743
lat            -0.158657  -0.138048 -0.056018  0.208792     0.054513
…                    …          …         …         …            …
separated       0.668481   1.000000  0.133244 -0.151824     0.094859
divorced        0.057364   0.133244  1.000000 -0.210203    -0.155328
bad_debt       -0.151008  -0.151824 -0.210203  1.000000    -0.005871
pop_density     0.212778   0.094859 -0.155328 -0.005871     1.000000
age_median     -0.190105  -0.116763  0.164205  0.058892    -0.198546

             age_median
COUNTYID      -0.063521
STATEID       -0.017172
zip_code      -0.126150
area_code     -0.017118
lat            0.008246
…                   …
separated     -0.116763
divorced       0.164205
bad_debt       0.058892
pop_density   -0.198546
age_median     1.000000

[74 rows x 74 columns]
```

[181]:
```python
cols=train.corr().nlargest(10,'hc_mortgage_mean')
```

[182]:
```python
plt.figure(figsize=(25,15))
sns.heatmap(cols,annot=True)
```

[182]:
```
<AxesSubplot:>
```



[184]:

```
#Data Pre-processing:
```

[185]:

```
pip install factor_analyzer
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: factor_analyzer in
/home/labsuser/.local/lib/python3.7/site-packages (0.4.1)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/site-packages
(from factor_analyzer) (1.4.1)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.7/site-
packages (from factor_analyzer) (1.0.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/site-packages
(from factor_analyzer) (1.21.5)
Requirement already satisfied: pre-commit in
/home/labsuser/.local/lib/python3.7/site-packages (from factor_analyzer)
(2.20.0)
Requirement already satisfied: pandas in /usr/local/lib/python3.7/site-packages
(from factor_analyzer) (1.1.5)
Requirement already satisfied: python-dateutil>=2.7.3 in
/usr/local/lib/python3.7/site-packages (from pandas->factor_analyzer) (2.8.1)
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.7/site-
packages (from pandas->factor_analyzer) (2019.3)
```

```
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.7/site-
packages (from pre-commit->factor_analyzer) (5.3.1)
Requirement already satisfied: nodeenv>=0.11.1 in
/home/labsuser/.local/lib/python3.7/site-packages (from pre-
commit->factor_analyzer) (1.7.0)
Requirement already satisfied: identify>=1.0.0 in
/home/labsuser/.local/lib/python3.7/site-packages (from pre-
commit->factor_analyzer) (2.5.9)
Requirement already satisfied: virtualenv>=20.0.8 in
/home/labsuser/.local/lib/python3.7/site-packages (from pre-
commit->factor_analyzer) (20.16.7)
Requirement already satisfied: toml in /usr/local/lib/python3.7/site-packages
(from pre-commit->factor_analyzer) (0.10.0)
Requirement already satisfied: importlib-metadata in
/usr/local/lib/python3.7/site-packages (from pre-commit->factor_analyzer)
(1.6.0)
Requirement already satisfied: cfgv>=2.0.0 in
/home/labsuser/.local/lib/python3.7/site-packages (from pre-
commit->factor_analyzer) (3.3.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in
/usr/local/lib/python3.7/site-packages (from scikit-learn->factor_analyzer)
(2.2.0)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/site-
packages (from scikit-learn->factor_analyzer) (0.14.1)
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/site-
packages (from nodeenv>=0.11.1->pre-commit->factor_analyzer) (41.2.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.7/site-
packages (from python-dateutil>=2.7.3->pandas->factor_analyzer) (1.14.0)
Requirement already satisfied: distlib<1,>=0.3.6 in
/home/labsuser/.local/lib/python3.7/site-packages (from virtualenv>=20.0.8->pre-
commit->factor_analyzer) (0.3.6)
Collecting importlib-metadata
  Downloading importlib_metadata-6.6.0-py3-none-any.whl (22 kB)
Requirement already satisfied: platformdirs<3,>=2.4 in
/home/labsuser/.local/lib/python3.7/site-packages (from virtualenv>=20.0.8->pre-
commit->factor_analyzer) (2.5.4)
Collecting filelock<4,>=3.4.1
  Downloading filelock-3.12.2-py3-none-any.whl (10 kB)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/site-
packages (from importlib-metadata->pre-commit->factor_analyzer) (3.1.0)
Requirement already satisfied: typing-extensions>=3.6.4 in
/usr/local/lib/python3.7/site-packages (from importlib-metadata->pre-
commit->factor_analyzer) (4.0.1)
Installing collected packages: importlib-metadata, filelock
```

```
ERROR: pip's dependency resolver does not currently take into account all
the packages that are installed. This behaviour is the source of the following
dependency conflicts.
konoha 4.6.5 requires overrides<4.0.0,>=3.0.0, which is not installed.
flair 0.8.1 requires more-itertools~=8.8.0, but you have more-itertools 8.2.0
which is incompatible.
konoha 4.6.5 requires importlib-metadata<4.0.0,>=3.7.0, but you have importlib-
metadata 6.6.0 which is incompatible.
konoha 4.6.5 requires requests<3.0.0,>=2.25.1, but you have requests 2.23.0
which is incompatible.
Successfully installed filelock-3.12.2 importlib-metadata-6.6.0
WARNING: You are using pip version 22.0.3; however, version 23.1.2 is
available.
You should consider upgrading via the '/usr/local/bin/python3.7 -m pip install
--upgrade pip' command.
Note: you may need to restart the kernel to use updated packages.
```

[186]:
```
from factor_analyzer import FactorAnalyzer
```

[187]:
```
fa=FactorAnalyzer(n_factors=5)
```

[189]:
```
fa.fit_transform(train.select_dtypes(exclude=('object','category')))
```

[189]:
```
array([[-0.41205343,  0.51294274,  0.87903004, -1.11001903,  0.35041992],
       [-1.04824274, -0.50174344, -0.39507676,  0.081311  ,  0.32595819],
       [ 0.11209985,  1.26467376,  0.76773891, -0.47930207, -0.36363692],
       …,
       [-0.02669751, -0.75106047,  0.77972285, -1.39880081,  0.03865004],
       [ 2.53195117,  3.0676096 ,  1.45490888, -0.07337594, -1.50506532],
       [-0.1992642 ,  0.01415226, -1.23527594,  0.25760531, -0.04155054]])
```

[190]:
```
#Data Modeling :
```

[191]:
```
train.columns
```

[191]:
```
Index(['COUNTYID', 'STATEID', 'state', 'state_ab', 'city', 'place', 'type',
       'primary', 'zip_code', 'area_code', 'lat', 'lng', 'ALand', 'AWater',
       'pop', 'male_pop', 'female_pop', 'rent_mean', 'rent_median',
       'rent_stdev', 'rent_sample_weight', 'rent_samples', 'rent_gt_10',
       'rent_gt_15', 'rent_gt_20', 'rent_gt_25', 'rent_gt_30', 'rent_gt_35',
```

```
 'rent_gt_40', 'rent_gt_50', 'universe_samples', 'used_samples',
 'hi_mean', 'hi_median', 'hi_stdev', 'hi_sample_weight', 'hi_samples',
 'family_mean', 'family_median', 'family_stdev', 'family_sample_weight',
 'family_samples', 'hc_mortgage_mean', 'hc_mortgage_median',
 'hc_mortgage_stdev', 'hc_mortgage_sample_weight', 'hc_mortgage_samples',
 'hc_mean', 'hc_median', 'hc_stdev', 'hc_samples', 'hc_sample_weight',
 'home_equity_second_mortgage', 'second_mortgage', 'home_equity', 'debt',
 'second_mortgage_cdf', 'home_equity_cdf', 'debt_cdf', 'hs_degree',
 'hs_degree_male', 'hs_degree_female', 'male_age_mean',
 'male_age_median', 'male_age_stdev', 'male_age_sample_weight',
 'male_age_samples', 'female_age_mean', 'female_age_median',
 'female_age_stdev', 'female_age_sample_weight', 'female_age_samples',
 'pct_own', 'married', 'married_snp', 'separated', 'divorced',
 'bad_debt', 'pop_density', 'age_median', 'pop_bins'],
dtype='object')
```

[193]:
```
train['type']
```

[193]:
```
UID
267822       City
246444       City
245683       City
279653      Urban
247218       City
            …
279212      Urban
277856    Borough
233000       City
287425       Town
265371       City
Name: type, Length: 27321, dtype: object
```

[194]:
```
# convert type column into numerical data
train.replace({'City':1,'Town':2,'CDP':3,'Village':4,'Borough':5,'Urban':
↪6},inplace=True)
```

[195]:
```
train['type'].value_counts()
```

[195]:
```
1    15237
2     3666
3     3658
4     3216
5     1226
6      318
Name: type, dtype: int64
```

[198]:
```
test.replace({'City':1,'Town':2,'CDP':3,'Village':4,'Borough':5,'Urban':
↪6},inplace=True)
```

[200]:
```
test['type'].value_counts()
```

[200]:
```
1    6481
2    1634
3    1558
4    1356
5     509
6     171
Name: type, dtype: int64
```

[202]:
```
train.columns
```

[202]:
```
Index(['COUNTYID', 'STATEID', 'state', 'state_ab', 'city', 'place', 'type',
       'primary', 'zip_code', 'area_code', 'lat', 'lng', 'ALand', 'AWater',
       'pop', 'male_pop', 'female_pop', 'rent_mean', 'rent_median',
       'rent_stdev', 'rent_sample_weight', 'rent_samples', 'rent_gt_10',
       'rent_gt_15', 'rent_gt_20', 'rent_gt_25', 'rent_gt_30', 'rent_gt_35',
       'rent_gt_40', 'rent_gt_50', 'universe_samples', 'used_samples',
       'hi_mean', 'hi_median', 'hi_stdev', 'hi_sample_weight', 'hi_samples',
       'family_mean', 'family_median', 'family_stdev', 'family_sample_weight',
       'family_samples', 'hc_mortgage_mean', 'hc_mortgage_median',
       'hc_mortgage_stdev', 'hc_mortgage_sample_weight', 'hc_mortgage_samples',
       'hc_mean', 'hc_median', 'hc_stdev', 'hc_samples', 'hc_sample_weight',
       'home_equity_second_mortgage', 'second_mortgage', 'home_equity', 'debt',
       'second_mortgage_cdf', 'home_equity_cdf', 'debt_cdf', 'hs_degree',
       'hs_degree_male', 'hs_degree_female', 'male_age_mean',
       'male_age_median', 'male_age_stdev', 'male_age_sample_weight',
       'male_age_samples', 'female_age_mean', 'female_age_median',
       'female_age_stdev', 'female_age_sample_weight', 'female_age_samples',
       'pct_own', 'married', 'married_snp', 'separated', 'divorced',
       'bad_debt', 'pop_density', 'age_median', 'pop_bins'],
      dtype='object')
```

[203]:
```
input_cols=['COUNTYID',␣
↪'STATEID','type','zip_code','pop','family_mean','second_mortgage',␣
↪'home_equity','debt','hs_degree','age_median','pct_own','married','separated',␣
↪'divorced']
```

[204]:
```
x_train=train[input_cols]
```

[205]:
```
x_train
```

[205]:
```
        COUNTYID  STATEID  type  zip_code    pop  family_mean  \
UID
```

```
267822         53         36      1      13346    5230    67994.14790
246444        141         18      1      46616    2633    50670.10337
245683         63         18      1      46122    6881    95262.51431
279653        127         72      6        927    2700    56401.68133
247218        161         20      1      66502    5637    54053.42396
…           …       …    …       …      …       …
279212         43         72      6        769    1847    20889.14617
277856         91         42      5      19422    4155   118896.06830
233000         87          8      1      80653    2829    88878.57034
287425        439         48      2      76034   11542   167148.83770
265371          3         32      1      89123    3726    54886.07827

        second_mortgage  home_equity     debt  hs_degree  age_median  pct_own  \
UID
267822          0.02077      0.08919  0.52963    0.89288   44.666665  0.79046
246444          0.02222      0.04274  0.60855    0.90487   34.791665  0.52483
245683          0.00000      0.09512  0.73484    0.94288   41.833330  0.85331
279653          0.01086      0.01086  0.52714    0.91500   49.750000  0.65037
247218          0.05426      0.05426  0.51938    1.00000   22.000000  0.13046
…                  …           …      …        …          …          …
279212          0.00000      0.00000  0.11694    0.60155   40.916670  0.60422
277856          0.02112      0.19641  0.65364    0.95737   39.166665  0.68072
233000          0.02024      0.07857  0.58095    0.93555   44.166665  0.78508
287425          0.07550      0.12556  0.65722    0.98540   45.041670  0.93970
265371          0.01412      0.18362  0.65537    0.87370   31.166665  0.27912

        married  separated  divorced
UID
267822  0.57851    0.01240   0.08770
246444  0.34886    0.01426   0.09030
245683  0.64745    0.01607   0.10657
279653  0.47257    0.02021   0.10106
247218  0.12356    0.00000   0.03109
…           …          …        …
279212  0.24603    0.02249   0.14683
277856  0.61127    0.02473   0.04888
233000  0.70451    0.00520   0.07712
287425  0.75503    0.00915   0.05261
265371  0.34426    0.03005   0.13320

[27321 rows x 15 columns]
```

[207]:
```
y_train=train['hc_mortgage_mean']
```

[208]:
```
y_train
```

[208]: UID
```
267822    1414.80295
246444     864.41390
245683    1506.06758
279653    1175.28642
247218    1192.58759
             …
279212     770.11560
277856    2210.84055
233000    1671.07908
287425    3074.83088
265371    1455.42340
Name: hc_mortgage_mean, Length: 27321, dtype: float64
```

[214]:
```
x_test=test[input_cols]
y_test=test['hc_mortgage_mean']
```

[211]:
```
from sklearn.preprocessing import StandardScaler
```

[213]:
```
sc=StandardScaler()
```

[216]:
```
x_train_scaled=sc.fit_transform(x_train)
x_test_scaled=sc.fit_transform(x_test)
```

[217]:
```
#apply linear regression model
from sklearn.linear_model import LinearRegression
linear_reg=LinearRegression()
linear_reg.fit(x_train_scaled,y_train)
y_pred=linear_reg.predict(x_test_scaled)
```

[218]:
```
y_pred
```

[218]:
```
array([ 874.67481013, 1597.10903054, 1086.41351981, …, 1915.00495942,
       1505.10480889, 1151.68011643])
```

[219]:
```
from sklearn.metrics import mean_squared_error,r2_score,accuracy_score
print('Mean Squared error',np.sqrt(mean_squared_error(y_test,y_pred)))
```

```
Mean Squared error 325.0919574748077
```

[220]:
```
#Run another model at State level. There are 52 states in USA.
```

[221]:
```
train['STATEID'].unique()
```

[221]:
```
array([36, 18, 72, 20,  1, 48, 45,  6,  5, 24, 17, 19, 47, 32, 22,  8, 44,
       28, 34, 41,  4, 12, 55, 42, 37, 51, 26, 39, 40, 13, 16, 46, 27, 29,
       53, 56,  9, 54, 21, 25, 11, 15, 30,  2, 33, 49, 50, 31, 38, 35, 23,
```

```
10])
```

[224]:

```
for i in [20,1,45]:
    print('state id-->',i)
    x_train_nation=train[train['COUNTYID']==i][input_cols]
    y_train_nation=train[train['COUNTYID']==i]['hc_mortgage_mean']
    x_test_nation=test[test['COUNTYID']==i][input_cols]
    y_test_nation=test[test['COUNTYID']==i]['hc_mortgage_mean']
    x_train_scaled_nation=sc.fit_transform(x_train_nation)
    x_test_scaled_nation=sc.fit_transform(x_test_nation)
    linear_reg.fit(x_train_scaled_nation,y_train_nation)
    yprd=linear_reg.predict(x_test_scaled_nation)
    print('root Mean Squared error',np.
 ↪sqrt(mean_squared_error(y_test_nation,yprd)))
    print('R2 score',r2_score(y_test_nation,yprd))
```

```
state id--> 20
root Mean Squared error 307.9718899931471
R2 score 0.6046603766461811
state id--> 1
root Mean Squared error 307.7896199248688
R2 score 0.8104850042868166
state id--> 45
root Mean Squared error 225.62754461084364
R2 score 0.7888730697076223
```

[ ]: