capstone_project_Real_estate

March 21, 2023

1 Data Import & preparatiions

[122]:
```
#Import data.

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[123]:
```
#import the dataset
df_train=pd.read_csv('train.csv')
```

[124]:
```
df_test=pd.read_csv('test.csv')
```

[125]:
```
df_train.shape
```

[125]:
```
(27321, 80)
```

[126]:
```
df_test.shape
```

[126]:
```
(11709, 80)
```

[127]:
```
df_train.head()
```

[127]:
```
      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]
```

[128]:
```
df_test.head()
```

[128]:
```
      UID  BLOCKID  SUMLEVEL  COUNTYID  STATEID         state state_ab  \
0  255504      NaN       140       163       26      Michigan       MI
1  252676      NaN       140         1       23         Maine       ME
2  276314      NaN       140        15       42  Pennsylvania       PA
3  248614      NaN       140       231       21      Kentucky       KY
4  286865      NaN       140       355       48         Texas       TX

             city                  place     type  … female_age_mean  \
0         Detroit  Dearborn Heights City      CDP  …        34.78682
1          Auburn            Auburn City     City  …        44.23451
2       Pine City              Millerton  Borough  …        41.62426
3      Monticello        Monticello City     City  …        44.81200
4  Corpus Christi                  Edroy     Town  …        40.66618

   female_age_median  female_age_stdev  female_age_sample_weight  \
0           33.75000          21.58531                 416.48097
1           46.66667          22.37036                 532.03505
2           44.50000          22.86213                 453.11959
3           48.00000          21.03155                 263.94320
4           42.66667          21.30900                 709.90829

   female_age_samples  pct_own  married  married_snp  separated  divorced
0              1938.0  0.70252  0.28217      0.05910    0.03813   0.14299
1              1950.0  0.85128  0.64221      0.02338    0.00000   0.13377
2              1879.0  0.81897  0.59961      0.01746    0.01358   0.10026
3              1081.0  0.84609  0.56953      0.05492    0.04694   0.12489
4              2956.0  0.79077  0.57620      0.01726    0.00588   0.16379
```

[5 rows x 80 columns]

[129]:
```
#train data
df_train.columns
```

[129]:
```
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')
```

[130]:
```
df_test.columns
```

[130]:
```
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')
```

[131]:
```python
#describe
df_train.describe()
```

[131]:
```
                 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     67.074017    -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      0.027488      0.129000
max        1.000000      0.714290      0.714290      1.000000

[8 rows x 74 columns]
```

[132]:
```
df_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
```

[ ]:

2 Figure out the primary key and look for the requirement of indexing.

[133]:
```python
#make UID as index
#df_train=df_train.set_index(keys=['UID'], inplace=True)
df_train.set_index(df_train['UID'],inplace=True)
df_train
```

[133]:
```
            UID  BLOCKID  SUMLEVEL  COUNTYID  STATEID         state state_ab  \
UID
267822   267822      NaN       140        53       36      New York       NY
246444   246444      NaN       140       141       18       Indiana       IN
245683   245683      NaN       140        63       18       Indiana       IN
279653   279653      NaN       140       127       72   Puerto Rico       PR
247218   247218      NaN       140       161       20        Kansas       KS
…           …      …        …        …        …          …       …
279212   279212      NaN       140        43       72   Puerto Rico       PR
277856   277856      NaN       140        91       42  Pennsylvania       PA
233000   233000      NaN       140        87        8      Colorado       CO
287425   287425      NaN       140       439       48         Texas       TX
265371   265371      NaN       140         3       32        Nevada       NV

               city              place     type  … female_age_mean  \
UID                                              …
267822     Hamilton           Hamilton     City  …        44.48629
246444   South Bend           Roseland     City  …        36.48391
245683     Danville           Danville     City  …        42.15810
279653     San Juan           Guaynabo    Urban  …        47.77526
247218    Manhattan     Manhattan City     City  …        24.17693
…             …                 …      …   …             …
279212        Coamo              Coamo    Urban  …        42.73154
277856    Blue Bell          Blue Bell  Borough  …        38.21269
233000      Weldona       Saddle Ridge     City  …        43.40218
287425  Colleyville   Colleyville City     Town  …        39.25921
265371    Las Vegas           Paradise     City  …        34.45345

        female_age_median  female_age_stdev  female_age_sample_weight  \
UID
```

```
267822          45.33333        22.51276                 685.33845
246444          37.58333        23.43353                 267.23367
245683          42.83333        23.94119                 707.01963
279653          50.58333        24.32015                 362.20193
247218          21.58333        11.10484                1854.48652
…                  …               …                       …
279212          40.16667        24.79821                 230.87898
277856          39.50000        21.84826                 496.20427
233000          46.33333        23.40858                 316.52078
287425          43.41667        21.36235                1373.94120
265371          29.83333        19.77208                 526.73261

        female_age_samples  pct_own  married  married_snp  separated  divorced
UID
267822              2618.0  0.79046  0.57851      0.01882    0.01240   0.08770
246444              1284.0  0.52483  0.34886      0.01426    0.01426   0.09030
245683              3238.0  0.85331  0.64745      0.02830    0.01607   0.10657
279653              1559.0  0.65037  0.47257      0.02021    0.02021   0.10106
247218              3051.0  0.13046  0.12356      0.00000    0.00000   0.03109
…                      …       …        …           …          …         …
279212               938.0  0.60422  0.24603      0.03042    0.02249   0.14683
277856              2039.0  0.68072  0.61127      0.05003    0.02473   0.04888
233000              1364.0  0.78508  0.70451      0.01386    0.00520   0.07712
287425              5815.0  0.93970  0.75503      0.02287    0.00915   0.05261
265371              1911.0  0.27912  0.34426      0.03825    0.03005   0.13320

[27321 rows x 80 columns]
```

[134]:
```python
df_test.set_index(df_test['UID'],inplace=True)
df_test
```

[134]:
```
           UID  BLOCKID  SUMLEVEL  COUNTYID  STATEID          state state_ab  \
UID
255504  255504      NaN       140       163       26       Michigan       MI
252676  252676      NaN       140         1       23          Maine       ME
276314  276314      NaN       140        15       42   Pennsylvania       PA
248614  248614      NaN       140       231       21       Kentucky       KY
286865  286865      NaN       140       355       48          Texas       TX
…          …       …         …         …        …           …        …
238088  238088      NaN       140       105       12        Florida       FL
242811  242811      NaN       140        31       17       Illinois       IL
250127  250127      NaN       140         9       25  Massachusetts       MA
241096  241096      NaN       140        27       19           Iowa       IA
287763  287763      NaN       140       453       48          Texas       TX

                    city                  place     type  … female_age_mean  \
UID                                                        …
```

```
255504         Detroit  Dearborn Heights City      CDP  …        34.78682
252676          Auburn             Auburn City     City  …        44.23451
276314       Pine City               Millerton  Borough  …        41.62426
248614      Monticello         Monticello City     City  …        44.81200
286865  Corpus Christi                   Edroy     Town  …        40.66618
…                  …                     …       …   …              …
238088        Lakeland         Crystal Springs     City  …        53.51255
242811         Chicago            Chicago City  Village  …        33.14169
250127        Lawrence       Methuen Town City     City  …        43.53905
241096         Carroll            Carroll City     City  …        45.63179
287763          Austin      Sunset Valley City     Town  …        35.99955

        female_age_median  female_age_stdev  female_age_sample_weight  \
UID
255504           33.75000          21.58531                 416.48097
252676           46.66667          22.37036                 532.03505
276314           44.50000          22.86213                 453.11959
248614           48.00000          21.03155                 263.94320
286865           42.66667          21.30900                 709.90829
…                     …               …                       …
238088           59.58333          23.23426                 699.33353
242811           32.83333          20.24698                 306.63915
250127           43.66667          23.17995                 900.13903
241096           48.16667          24.84209                 693.82905
287763           35.41667          20.68049                 559.30291

        female_age_samples  pct_own  married  married_snp  separated  divorced
UID
255504              1938.0  0.70252  0.28217      0.05910    0.03813   0.14299
252676              1950.0  0.85128  0.64221      0.02338    0.00000   0.13377
276314              1879.0  0.81897  0.59961      0.01746    0.01358   0.10026
248614              1081.0  0.84609  0.56953      0.05492    0.04694   0.12489
286865              2956.0  0.79077  0.57620      0.01726    0.00588   0.16379
…                      …       …       …          …          …         …
238088              2914.0  0.93121  0.65969      0.02135    0.02135   0.08780
242811              1191.0  0.33122  0.42882      0.07781    0.02829   0.05305
250127              3723.0  0.84372  0.50269      0.00108    0.00108   0.07294
241096              3213.0  0.83330  0.66699      0.02738    0.00000   0.04694
287763              2047.0  0.52587  0.51922      0.08066    0.02520   0.10586

[11709 rows x 80 columns]
```

Gauge the fill rate of the variables and devise plans for missing value treatment. Please explain explicitly the reason for the treatment chosen for each variable.

[135]:
```
df_train.isnull().sum().any()
```

[135]: True

[136]:
```
df_test.isnull().sum().any()
```

[136]: True

[137]:
```
#print only the columns in which we have missing values
df_test.isnull().sum()[df_test.isnull().sum()>0]
```

[137]:
```
BLOCKID                        11709
rent_mean                        148
rent_median                      148
rent_stdev                       148
rent_sample_weight               148
rent_samples                     148
rent_gt_10                       149
rent_gt_15                       149
rent_gt_20                       149
rent_gt_25                       149
rent_gt_30                       149
rent_gt_35                       149
rent_gt_40                       149
rent_gt_50                       149
hi_mean                          122
hi_median                        122
hi_stdev                         122
hi_sample_weight                 122
hi_samples                       122
family_mean                      136
family_median                    136
family_stdev                     136
family_sample_weight             136
family_samples                   136
hc_mortgage_mean                 268
hc_mortgage_median               268
hc_mortgage_stdev                268
hc_mortgage_sample_weight        268
hc_mortgage_samples              268
hc_mean                          290
hc_median                        290
hc_stdev                         290
hc_samples                       290
hc_sample_weight                 290
home_equity_second_mortgage      220
second_mortgage                  220
home_equity                      220
debt                             220
```

```
second_mortgage_cdf            220
home_equity_cdf                220
debt_cdf                       220
hs_degree                       85
hs_degree_male                  89
hs_degree_female               105
male_age_mean                   84
male_age_median                 84
male_age_stdev                  84
male_age_sample_weight          84
male_age_samples                84
female_age_mean                 96
female_age_median               96
female_age_stdev                96
female_age_sample_weight        96
female_age_samples              96
pct_own                        122
married                         84
married_snp                     84
separated                       84
divorced                        84
dtype: int64
```

[138]:
```
df_test.isnull().sum()[df_test.isnull().sum()>0].shape
```

[138]:
```
(59,)
```

[139]:
```
df_train.isnull().sum()[df_train.isnull().sum()>0]
```

[139]:
```
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
```

[140]:
```
df_train.isnull().sum()[df_train.isnull().sum()>0].shape
```

[140]:
```
(59,)
```

[141]:
```
#calculate % of missing values in each col
percent_train=df_train.isnull().sum()/len(df_train)*100
percent_train
```

[141]:
```
UID              0.000000
BLOCKID        100.000000
SUMLEVEL         0.000000
COUNTYID         0.000000
STATEID          0.000000
                   …
pct_own          0.980930
married          0.699096
married_snp      0.699096
separated        0.699096
divorced         0.699096
Length: 80, dtype: float64
```

[ ]:

[142]:
```
df_percent_train=pd.DataFrame(percent_train,columns=['percentage of missing␣
 ↪values'])
df_percent_train
```

[142]:
```
             percentage of missing values
UID                              0.000000
BLOCKID                        100.000000
SUMLEVEL                         0.000000
COUNTYID                         0.000000
STATEID                          0.000000
…                                     …
pct_own                          0.980930
married                          0.699096
married_snp                      0.699096
separated                        0.699096
divorced                         0.699096

[80 rows x 1 columns]
```

[143]:
```
df_percent_train.sort_values(by=['percentage of missing␣
 ↪values'],inplace=True,ascending=False)
#df_percent_train.sort_values(by=['Percentage of Missing␣
 ↪values'],inplace=True,ascending=False)
df_percent_train
```

[143]:
```
                  percentage of missing values
BLOCKID                             100.000000
hc_median                             2.196113
```

```
hc_sample_weight                    2.196113
hc_samples                          2.196113
hc_stdev                            2.196113
…                                        …
AWater                              0.000000
pop                                 0.000000
male_pop                            0.000000
female_pop                          0.000000
UID                                 0.000000

[80 rows x 1 columns]
```

[144]:
```
#calculate % of missing values in test data
percent_test=df_test.isnull().sum()/len(df_test)*100
df_percent_test=pd.DataFrame(percent_test,columns=['percentage of missing␣
 ↪values'])
df_percent_test
```

[144]:
```
             percentage of missing values
UID                          0.000000
BLOCKID                    100.000000
SUMLEVEL                     0.000000
COUNTYID                     0.000000
STATEID                      0.000000
…                                 …
pct_own                      1.041934
married                      0.717397
married_snp                  0.717397
separated                    0.717397
divorced                     0.717397

[80 rows x 1 columns]
```

[145]:
```
df_percent_test.sort_values(by=['percentage of missing␣
 ↪values'],inplace=True,ascending=False)
#df_percent_train.sort_values(by=['Percentage of Missing␣
 ↪values'],inplace=True,ascending=False)
df_percent_test
```

[145]:
```
                  percentage of missing values
BLOCKID                             100.000000
hc_sample_weight                      2.476727
hc_samples                            2.476727
hc_stdev                              2.476727
hc_median                             2.476727
…                                        …
AWater                                0.000000
```

```
pop                                      0.000000
male_pop                                 0.000000
female_pop                               0.000000
UID                                      0.000000

[80 rows x 1 columns]
```

[146]:
```python
#drop the BlockID, sumlevel
df_train.drop(columns=['UID',"BLOCKID",'SUMLEVEL'],inplace=True)
```

[147]:
```python
df_test.drop(columns=['UID',"BLOCKID",'SUMLEVEL'],inplace=True)
```

[148]:
```python
#columns in train data which are missing values
missing_values_train=[]
for col in df_train.columns:
    if df_train[col].isnull().sum()!=0:
        missing_values_train.append(col)
```

[149]:
```python
missing_values_train
```

[149]:
```
['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',
 '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']
```

[150]:
```
#columns in train data which are missing values
missing_values_test=[]
for col in df_test.columns:
    if df_test[col].isnull().sum()!=0:
        missing_values_test.append(col)
```

[151]:
```
missing_values_test
```

[151]:
```
['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',
'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']
```

fill the missing values with mean

[152]:
```
for col in df_train.columns:
 if col in (missing_values_train):
  df_train[col].replace(np.nan,df_train[col].mean(),inplace=True)
```

[153]:
```
for col in df_test.columns:
 if col in (missing_values_test):
  df_test[col].replace(np.nan,df_test[col].mean(),inplace=True)
```

[154]:
```
df_train.isnull().sum().any()
```

[154]:
```
False
```

[155]:
```
df_test.isnull().sum().any()
```

[155]:
```
False
```

2.1 Exploratory Data Analysis

[156]:
```
!pip install pandasql
```

```
Requirement already satisfied: pandasql in c:\users\shibn\anaconda3\lib\site-
packages (0.7.3)
Requirement already satisfied: sqlalchemy in c:\users\shibn\anaconda3\lib\site-
packages (from pandasql) (1.4.32)
Requirement already satisfied: numpy in c:\users\shibn\anaconda3\lib\site-
packages (from pandasql) (1.21.5)
Requirement already satisfied: pandas in c:\users\shibn\anaconda3\lib\site-
packages (from pandasql) (1.4.2)
Requirement already satisfied: python-dateutil>=2.8.1 in
c:\users\shibn\anaconda3\lib\site-packages (from pandas->pandasql) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in
c:\users\shibn\anaconda3\lib\site-packages (from pandas->pandasql) (2021.3)
Requirement already satisfied: six>=1.5 in c:\users\shibn\anaconda3\lib\site-
packages (from python-dateutil>=2.8.1->pandas->pandasql) (1.16.0)
Requirement already satisfied: greenlet!=0.4.17 in
c:\users\shibn\anaconda3\lib\site-packages (from sqlalchemy->pandasql) (1.1.1)
```

[157]:
```
df_train.columns
```

[157]:
```
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')
```

[158]:
```python
from pandasql import sqldf
q1="select place,pct_own,second_mortgage,lat,lng from df_train where pct_own>0.
↪10 and second_mortgage<0.5 order by second_mortgage DESC LIMIT 2500;"
```

[159]:
```python
query_fun=lambda q:sqldf(q,globals())
df_train_loc=query_fun(q1)
```

[160]:
```python
df_train_loc
```

[160]:

	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	33.983204	-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]

Bad Debt = second_mortgage + home_equity - home_equity_second_mortgage

[161]:
```python
df_train['bad_debt']=df_train['second_mortgage']+df_train['home_equity']-df_train['home_equity
```

[162]:
```
df_train['bad_debt']
```

[162]:
```
UID
267822    0.09408
246444    0.04274
245683    0.09512
279653    0.01086
247218    0.05426
            …
279212    0.00000
277856    0.20908
233000    0.07857
287425    0.14305
265371    0.18362
Name: bad_debt, Length: 27321, dtype: float64
```

Create Box and whisker plot and analyze the distribution for 2nd mortgage, home equity, good debt, and bad debt for different cities

[163]:
```
df_train['city']
```

[163]:
```
UID
267822       Hamilton
246444     South Bend
245683       Danville
279653       San Juan
247218      Manhattan
                …
279212          Coamo
277856      Blue Bell
233000        Weldona
287425    Colleyville
265371      Las Vegas
Name: city, Length: 27321, dtype: object
```

[164]:
```
df_ham=df_train.loc[df_train['city']=='Hamilton']
df_Man=df_train.loc[df_train['city']=='Manhattan']
```

[165]:
```
df_box_city=pd.concat([df_ham,df_Man])
```

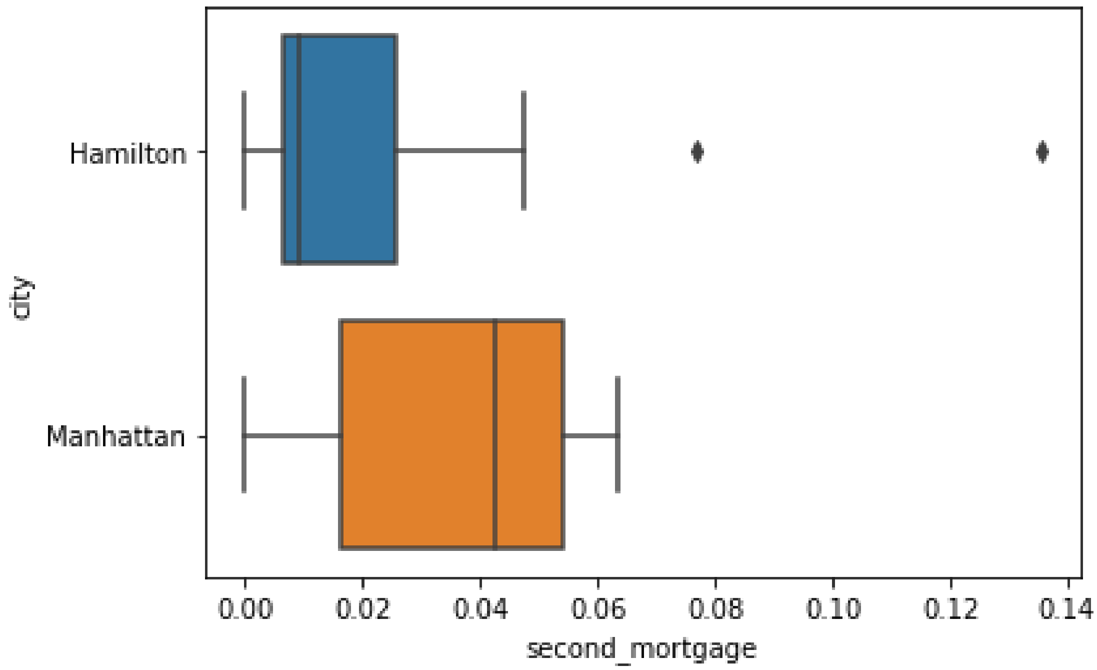
[166]:
```
df_box_city.head()
```

[166]:
```
        COUNTYID  STATEID        state state_ab      city          place  \
UID
267822        53       36     New York       NY  Hamilton       Hamilton
263797        21       34   New Jersey       NJ  Hamilton      Yardville
270979        17       39         Ohio       OH  Hamilton  Hamilton City
```

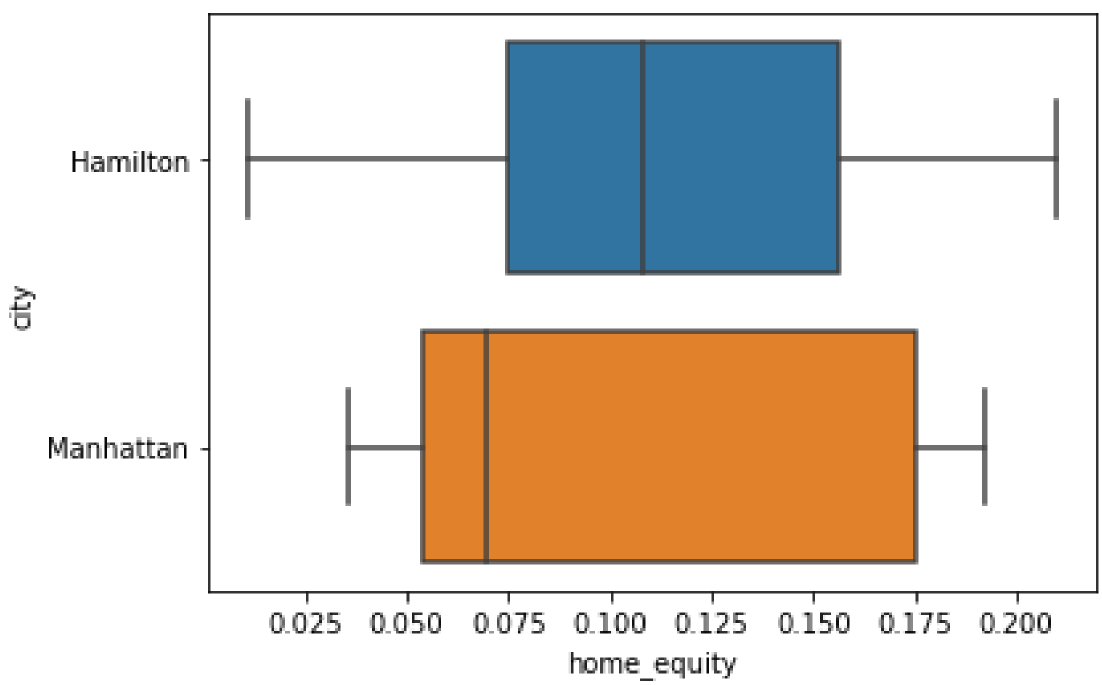
```
259028        95       28  Mississippi       MS  Hamilton       Hamilton
270984        17       39         Ohio       OH  Hamilton      New Miami

           type primary  zip_code  area_code  …  female_age_median  \
UID                                           …
267822     City   tract     13346        315  …           45.33333
263797     City   tract      8610        609  …           55.00000
270979  Village   tract     45015        513  …           31.66667
259028      CDP   tract     39746        662  …           35.91667
270984  Village   tract     45013        513  …           52.33333

        female_age_stdev  female_age_sample_weight  female_age_samples  \
UID
267822          22.51276                 685.33845              2618.0
263797          24.05831                 732.58443              3124.0
270979          22.66500                 565.32725              2528.0
259028          22.79602                 483.01311              1954.0
270984          24.55724                 682.81171              2912.0

        pct_own  married  married_snp  separated  divorced  bad_debt
UID
267822  0.79046  0.57851      0.01882    0.01240   0.08770   0.09408
263797  0.64400  0.56377      0.01980    0.00990   0.04892   0.18071
270979  0.61278  0.47397      0.04419    0.02663   0.13741   0.15005
259028  0.83241  0.58678      0.01052    0.00000   0.11721   0.02130
270984  0.63194  0.55697      0.01322    0.00000   0.15209   0.15651

[5 rows x 78 columns]
```

[167]:

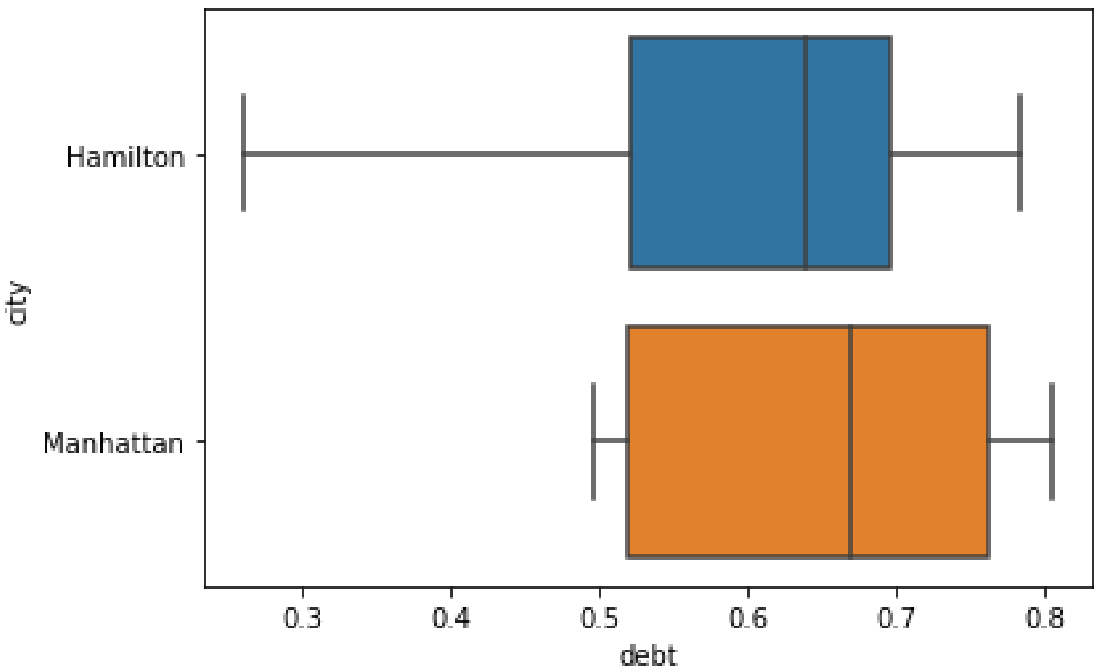
```
#create a boxplot city &second mortgage
sns.boxplot(data=df_box_city,x="second_mortgage",y='city')
plt.show()
```



[168]:
```
#create a boxplot
##create a box plot with city & home_equity
sns.boxplot(data=df_box_city,x='home_equity',y='city')
plt.show()
```

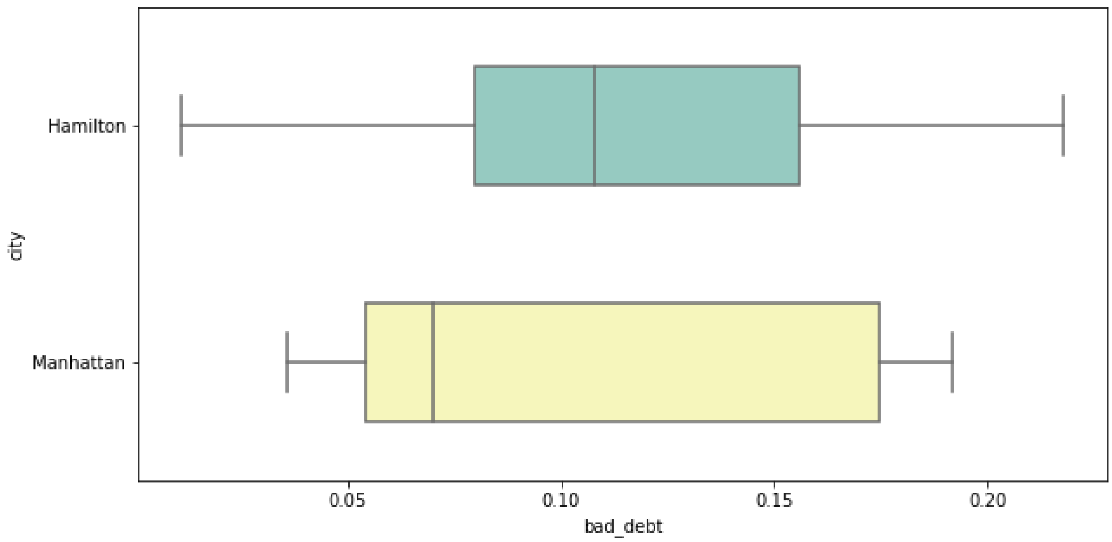


[169]:
```
##create a box plot with city & good_debt
sns.boxplot(data=df_box_city,x='debt',y='city')
plt.show()
```



[170]:
```
##create a box plot with city & bad debt
plt.figure(figsize=(10,5))
sns.boxplot(data=df_box_city,x='bad_debt', y='city',width=0.5,palette="Set3")
plt.show
```

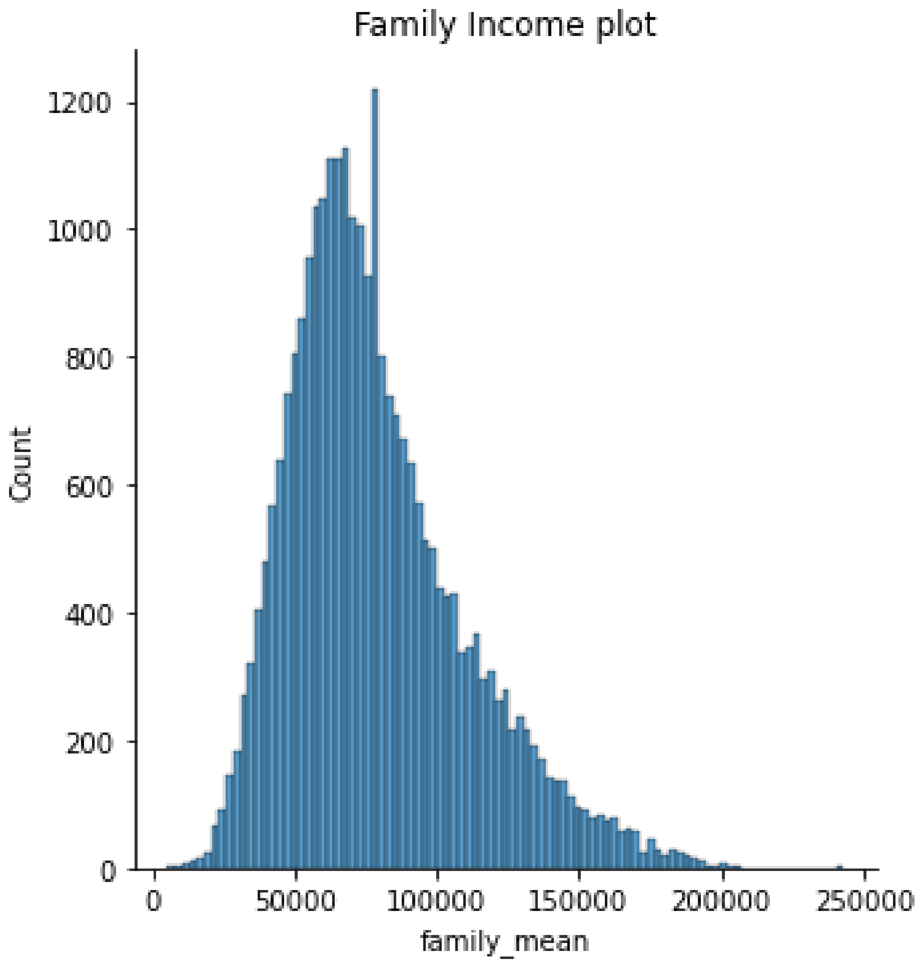
[170]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Create a collated income distribution chart for family income, house hold income, and remaining income

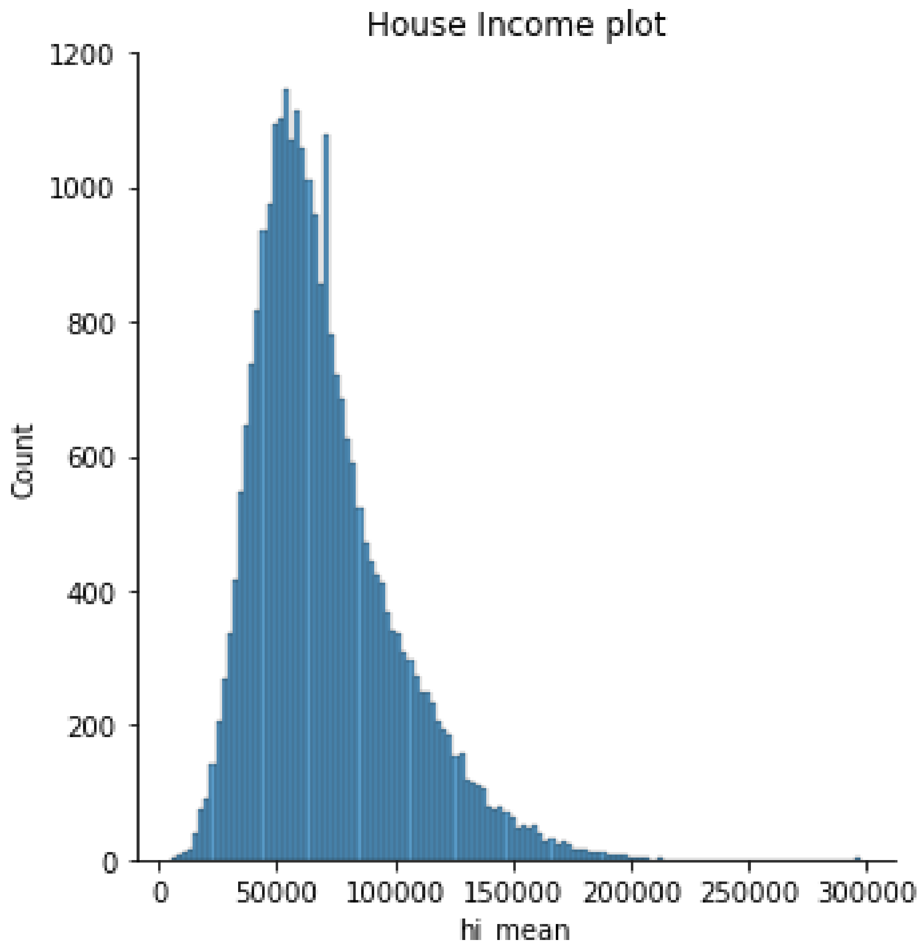
[171]:
```
sns.displot(df_train["family_mean"])
plt.title("Family Income plot")
```

[171]: Text(0.5, 1.0, 'Family Income plot')



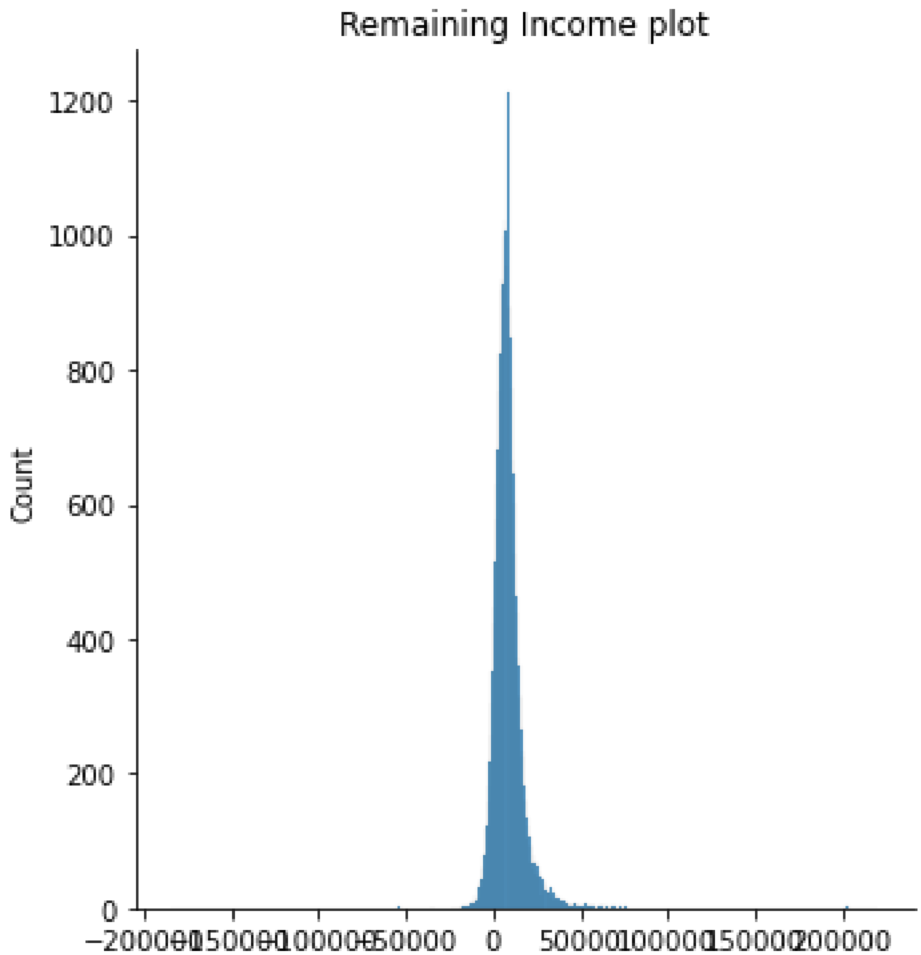
[172]:
```
sns.displot(df_train["hi_mean"])
plt.title("House Income plot")
```

[172]: Text(0.5, 1.0, 'House Income plot')



[173]:
```
sns.displot(df_train["family_mean"]-df_train["hi_mean"])
plt.title("Remaining Income plot")
```

[173]:
```
Text(0.5, 1.0, 'Remaining Income plot')
```



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

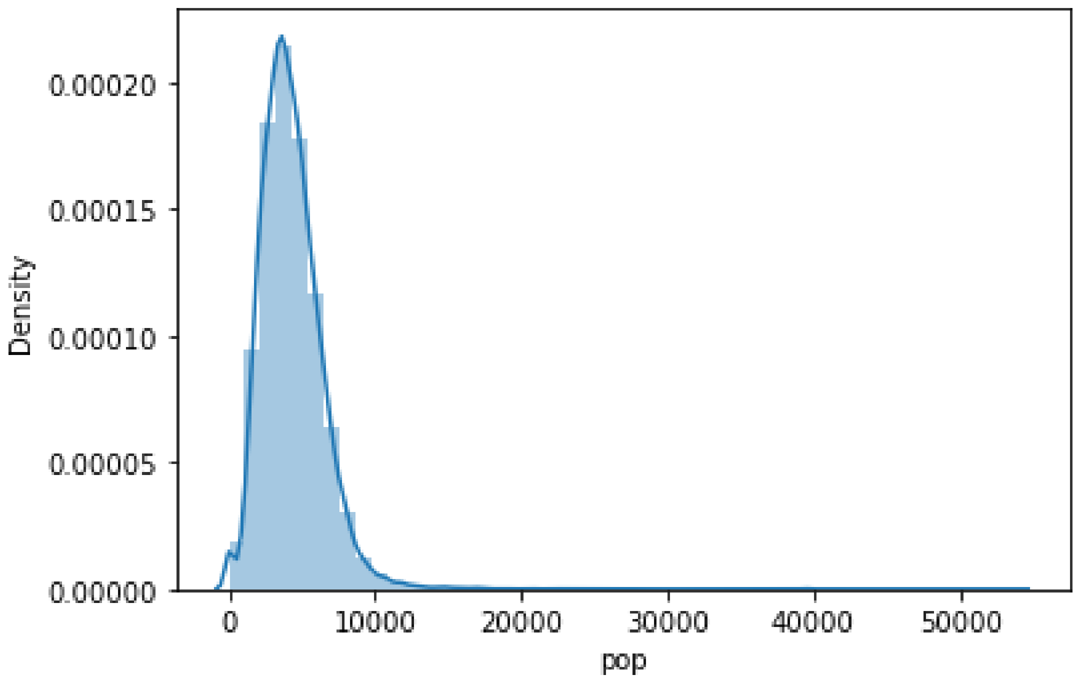
[174]:
```
df_train.columns
```

[174]:
```
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')
```

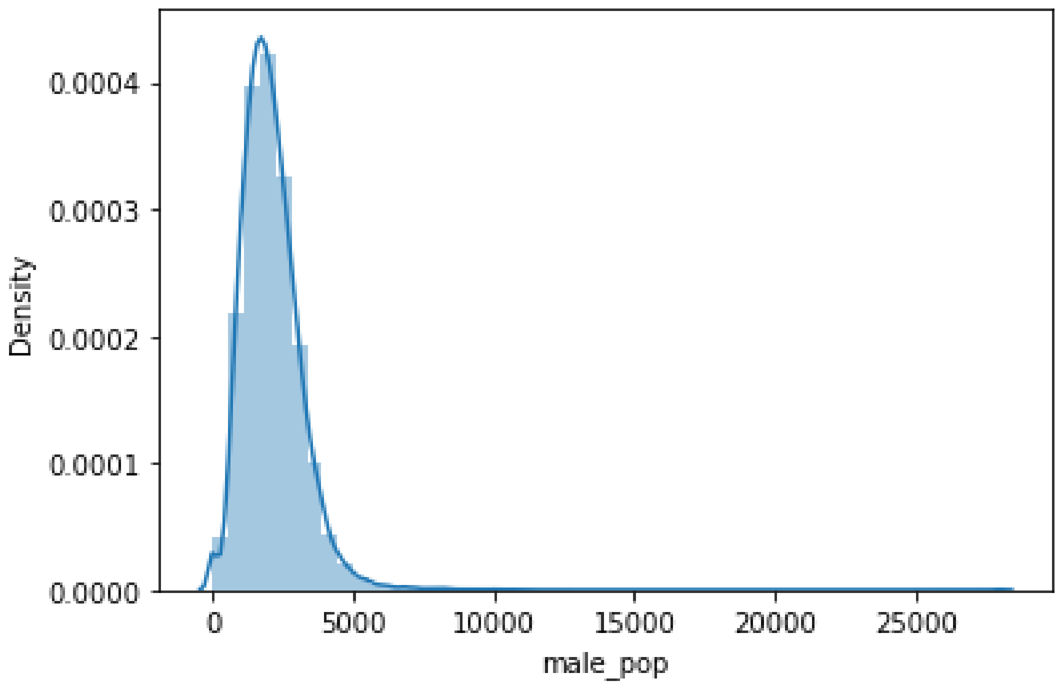
[175]:
```
sns.distplot(df_train['pop'])
```

[175]:
```
<AxesSubplot:xlabel='pop', ylabel='Density'>
```



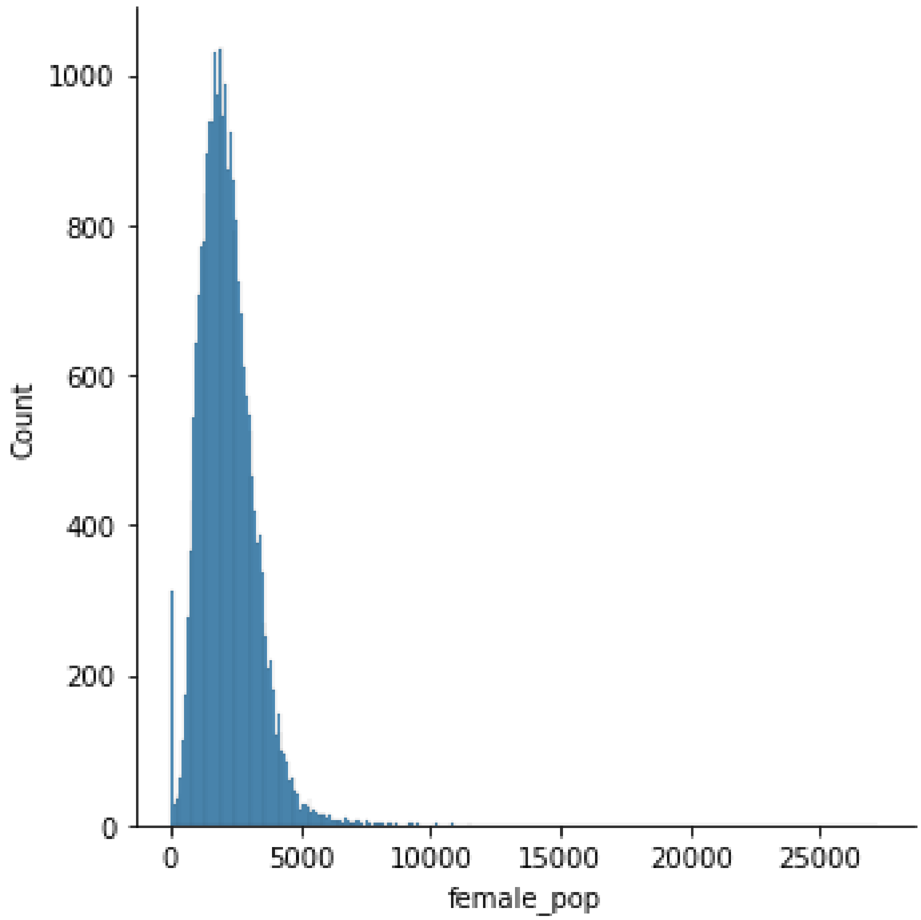
[176]:
```
sns.distplot(df_train['male_pop'])
```

[176]:
```
<AxesSubplot:xlabel='male_pop', ylabel='Density'>
```



[177]:
```
sns.displot(df_train['female_pop'])
```

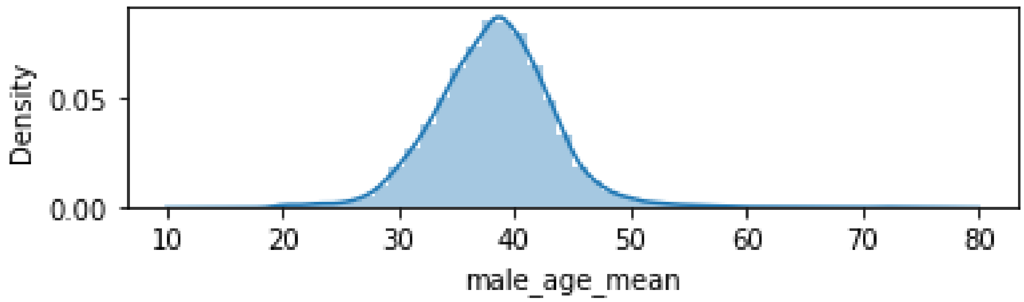
[177]: <seaborn.axisgrid.FacetGrid at 0x1bde6277eb0>

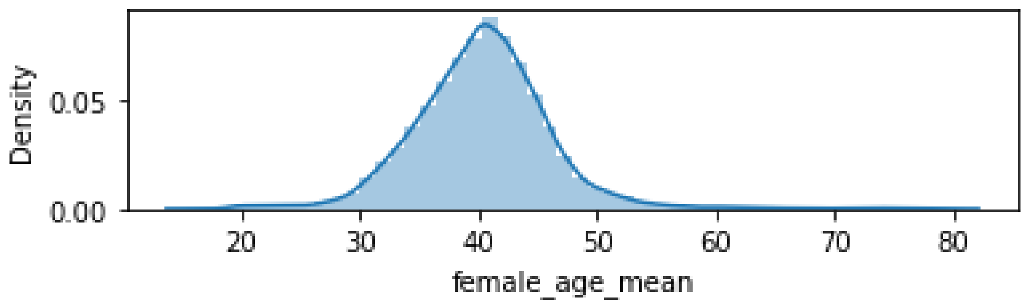


[178]:
```
import warnings
warnings.filterwarnings('ignore')
```

[179]:
```
fig,(ax1,ax2)=plt.subplots(2,1)
plt.subplots_adjust(wspace=0.8,hspace=0.9)
sns.distplot(df_train['male_age_mean'],ax=ax1)
sns.distplot(df_train['female_age_mean'],ax=ax2)
```

[179]:
```
<AxesSubplot:xlabel='female_age_mean', ylabel='Density'>
```





Use pop and Aland variables to create a new field -population density

[180]:
```
df_train['pop_density']=df_train['pop']/df_train['ALand']
```

[181]:
```
df_train['pop_density']
```

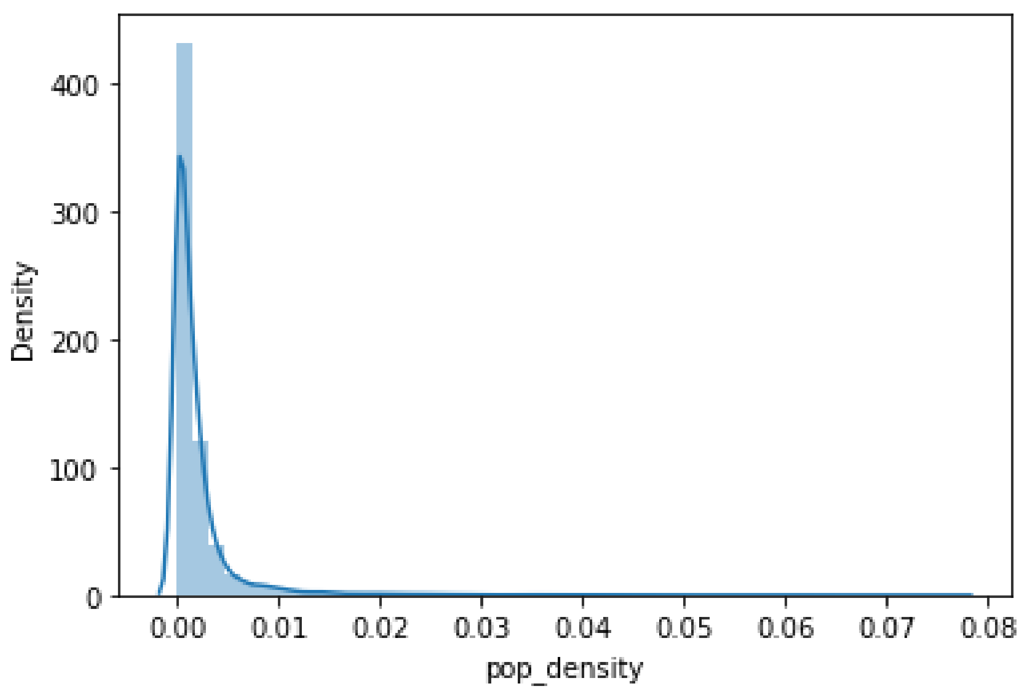
[181]:
```
UID
267822    0.000026
246444    0.001687
245683    0.000099
279653    0.002442
247218    0.002207
            …
279212    0.002650
277856    0.000818
233000    0.000002
287425    0.000619
265371    0.000478
Name: pop_density, Length: 27321, dtype: float64
```

[182]:
```
df_test['pop_density']=df_test['pop']/df_test['ALand']
```

[183]:
```
df_test['pop_density']
```

[183]:
```
UID
255504    0.001260
252676    0.000257
276314    0.000015
248614    0.000005
286865    0.000452
            …
238088    0.000061
242811    0.008241
250127    0.001415
241096    0.000537
287763    0.002069
Name: pop_density, Length: 11709, dtype: float64
```

[184]:
```
#check population density
sns.distplot(df_train['pop_density'])
plt.show()
```



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

[185]:
```
df_train['age_median']=(df_train['male_age_mean']+df_train['female_age_mean'])/2
```

[186]:
```
df_train['age_median']
```

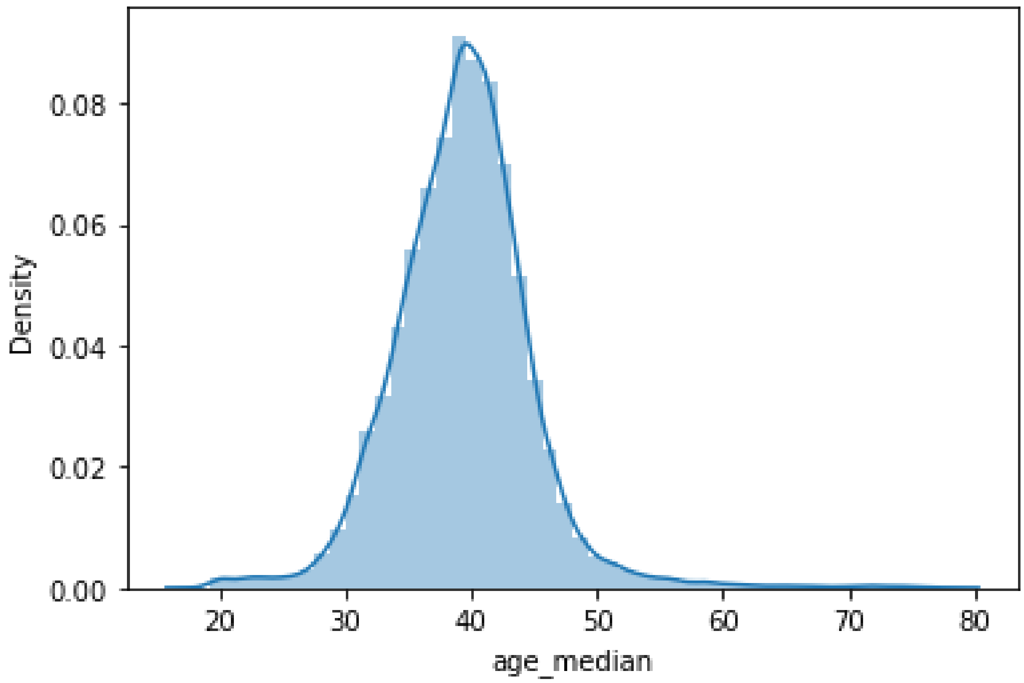
[186]:
```
UID
267822    43.486015
246444    35.665595
245683    40.769820
279653    48.211375
247218    25.126130
             …
279212    42.435825
277856    37.983820
233000    41.706760
287425    40.006650
265371    34.631955
Name: age_median, Length: 27321, dtype: float64
```

[187]:
```
df_test['age_median']=(df_test['male_age_mean']+df_test['female_age_mean'])/2
```

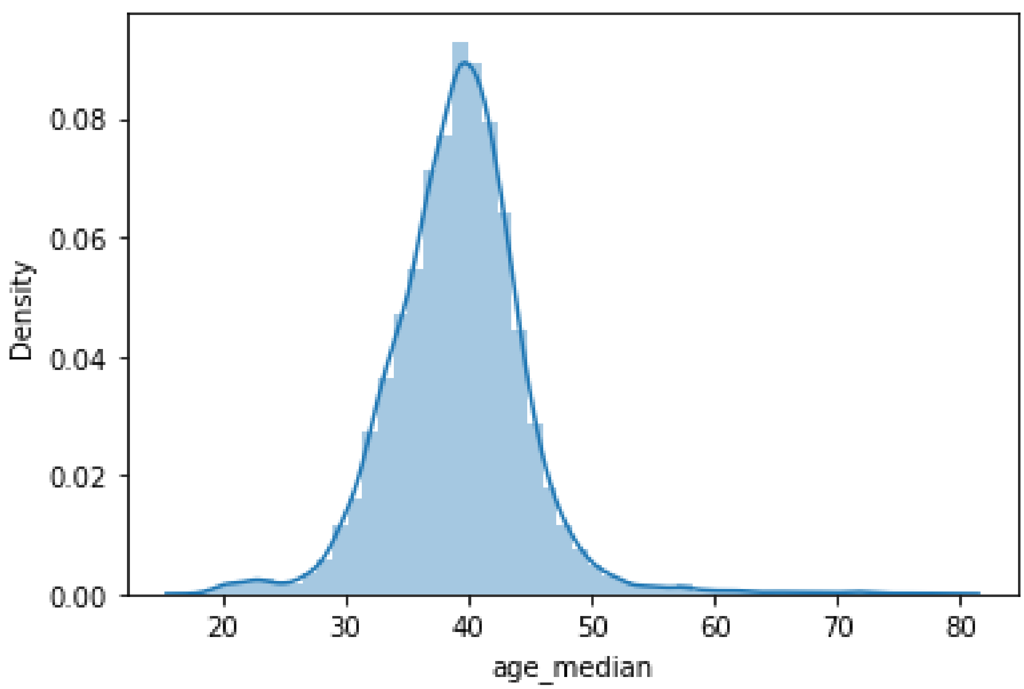
[188]:
```
df_test['age_median']
```

[188]:
```
UID
255504    34.079065
252676    44.060655
276314    40.720435
248614    43.314190
286865    41.399595
             …
238088    52.273950
242811    33.041570
250127    39.698240
241096    42.406990
287763    35.781795
Name: age_median, Length: 11709, dtype: float64
```

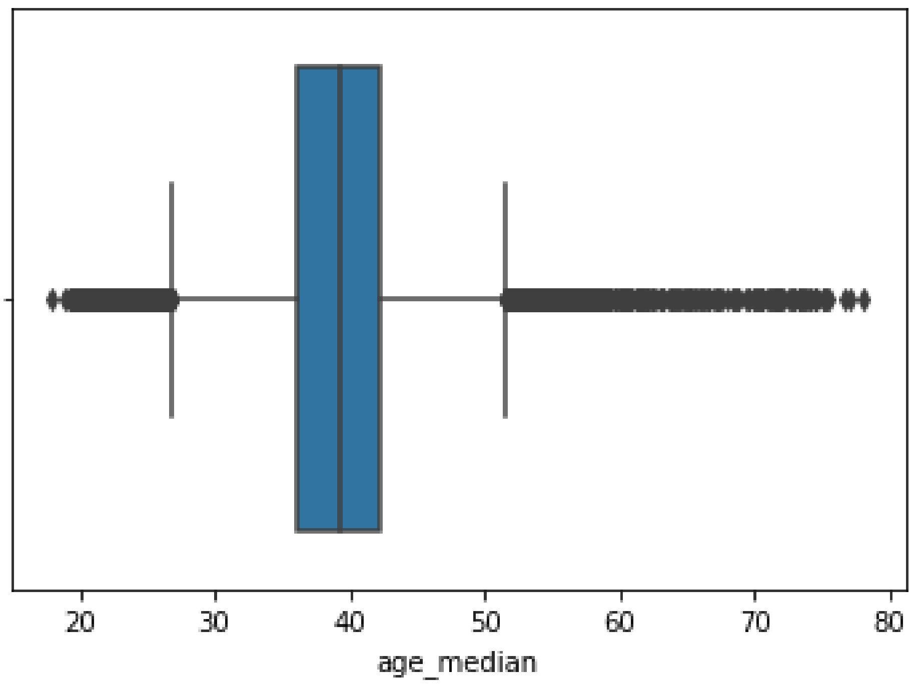
[189]:
```
# visualize the age median
sns.distplot(df_train['age_median'])
plt.show()
```



[190]:
```
sns.distplot(df_test['age_median'])
plt.show()
```



[191]:
```
# visualize the age_median
sns.boxplot(df_train['age_median'])
plt.show()
```



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

[192]:
```
df_train['pop'].head()
```

[192]:
```
UID
267822    5230
246444    2633
245683    6881
279653    2700
247218    5637
Name: pop, dtype: int64
```

[193]:
```
# apply function
def func(num):
    if num<7000:
        return 'low'
```

[194]:
```
df_train['pop_bin']=df_train['pop'].apply(func)
```

[195]:
```
df_train['pop_bin']
```

[195]:
```
UID
267822     low
246444     low
245683     low
279653     low
247218     low
          …
279212     low
277856     low
233000     low
287425    None
265371     low
Name: pop_bin, Length: 27321, dtype: object
```

[196]:
```
df_train['pop_bin'].value_counts()
```

[196]:
```
low    24883
Name: pop_bin, dtype: int64
```

[197]:
```
df_train['pop_binss']=pd.cut(df_train['pop'],bins=5,labels=['very␣
↪low','low','medium','high','very high'])
```

[198]:
```
df_train['pop_binss'].value_counts()
```

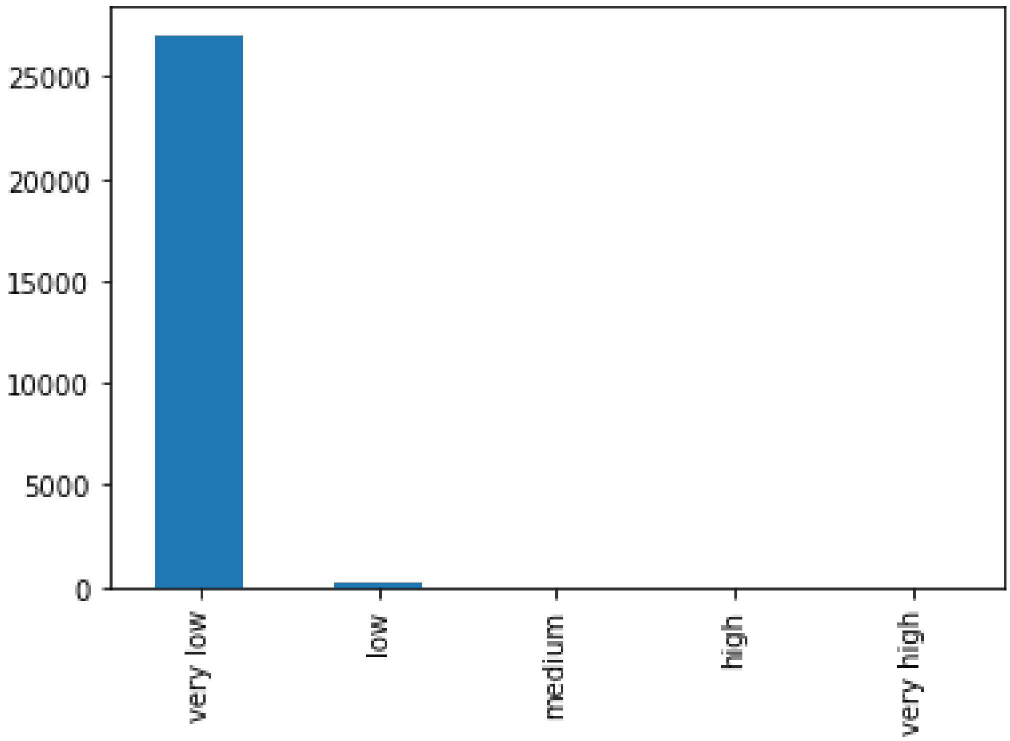
[198]:
```
very low     27058
low            246
medium           9
high             7
very high        1
Name: pop_binss, dtype: int64
```

[199]:
```
df_train[['pop','pop_bin']].head()
```

[199]:
```
         pop pop_bin
UID
267822  5230     low
246444  2633     low
245683  6881     low
279653  2700     low
247218  5637     low
```

[200]:
```
df_train['pop_binss'].value_counts().plot(kind='bar')
```

[200]:
```
<AxesSubplot:>
```



[ ]:
```

```

[201]:
```
df_train.columns
```

[201]:
```
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_bin', 'pop_binss'],
      dtype='object')
```

Analyze the married, separated, and divorced population for these population brackets

[202]:
```
df_train.groupby(by='pop_binss')[['married','separated', 'divorced']].count()
```

[202]:

pop_binss	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

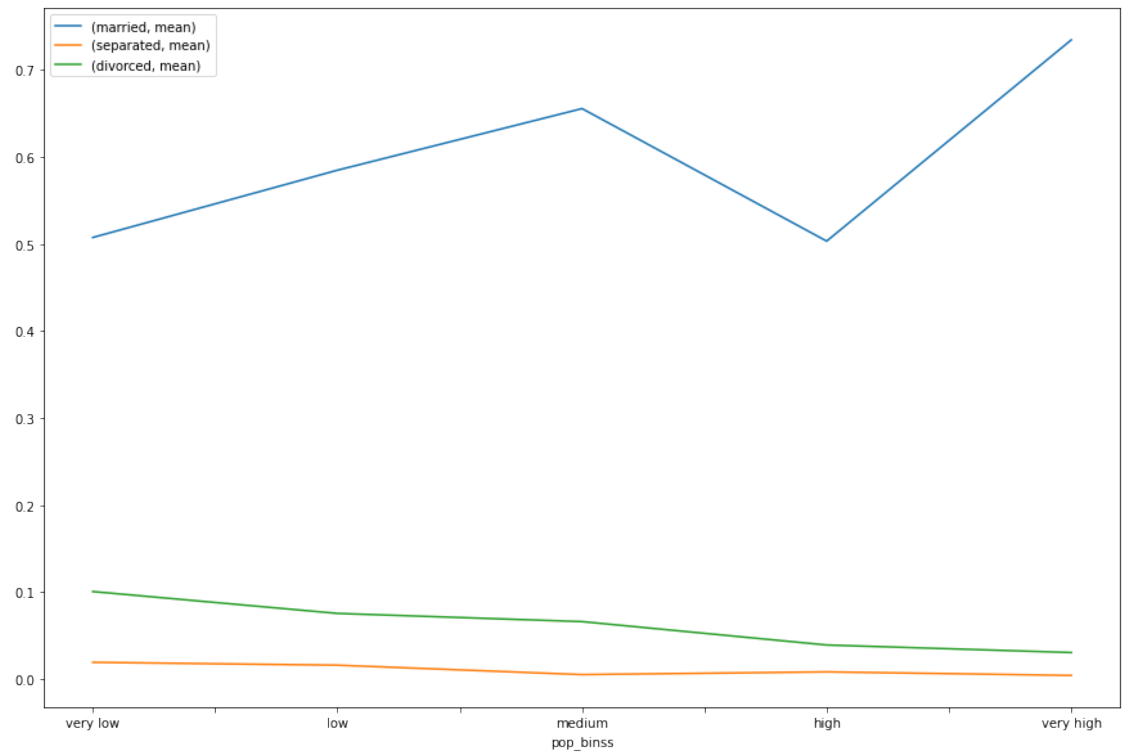
[203]:
```
df_train.groupby(by='pop_binss')[['married','separated', 'divorced']].
 ↪agg(['sum','mean','median','count'])
```

[203]:

	married				separated				divorced			
	sum	mean	median	count	sum	mean	median	count	sum	mean	median	count
pop_binss												
very low	13733.22489	0.507548	0.524680	27058	517.52126	0.019126	0.013650	27058	2719.430721	0.100504	0.096020	27058
low	143.88385	0.584894	0.593135	246	3.89480	0.015833	0.011195	246	18.535600	0.075348	0.070045	246
medium	5.90163	0.655737	0.618710	9	0.04503	0.005003	0.004120	9	0.593340	0.065927	0.064890	9
high	3.52351	0.503359	0.335660	7	0.05699	0.008141	0.002500	7	0.273210	0.039030	0.010320	7
very high	0.73474	0.734740	0.734740	1	0.00405	0.004050	0.004050	1	0.030360	0.030360	0.030360	1

[204]:
```
df_train.groupby(by='pop_binss')[['married','separated', 'divorced']].
 ↪agg(['mean']).plot(figsize=(15,10))
plt.legend(loc='best')
```

[204]: <matplotlib.legend.Legend at 0x1bdda228e20>



Please detail your observations for rent as a percentage of income at an overall level, and for different states. Perform correlation analysis for all the relevant variables by creating a heatmap. Describe your findings.

[205]:
```
rent_state_mean =df_train.groupby(by='state')['rent_mean'].agg(["mean"])
```

[206]:
```
rent_state_mean
```

[206]:

state	mean
Alabama	774.004927
Alaska	1185.763570
Arizona	1097.753511
Arkansas	720.918575
California	1471.133857
Colorado	1198.191514
Connecticut	1317.100534
Delaware	1127.309811
District of Columbia	1417.097934
Florida	1141.758549
Georgia	964.575973
Hawaii	1710.629412

```
Idaho                    800.486650
Illinois                1034.887921
Indiana                  810.910355
Iowa                     737.246152
Kansas                   831.215856
Kentucky                 742.199763
Louisiana                846.375506
Maine                    829.941899
Maryland                1412.009565
Massachusetts           1211.811159
Michigan                 928.123200
Minnesota                957.376502
Mississippi              738.111770
Missouri                 829.011192
Montana                  776.337306
Nebraska                 835.165893
Nevada                  1128.641766
New Hampshire           1083.090073
New Jersey              1379.709933
New Mexico               853.611858
New York                1248.850743
North Carolina           885.593430
North Dakota             771.423137
Ohio                     820.004760
Oklahoma                 777.702422
Oregon                  1024.616948
Pennsylvania             949.580140
Puerto Rico              550.079459
Rhode Island            1039.482069
South Carolina           859.919160
South Dakota             685.325569
Tennessee                856.649930
Texas                    977.074993
Utah                    1068.930520
Vermont                  937.119939
Virginia                1305.707687
Washington              1126.649264
West Virginia            667.193267
Wisconsin                841.670190
Wyoming                  861.395327
```

[207]:
```
income_state_mean =df_train.groupby(by='state')['family_mean'].agg(["mean"])
```

[208]:
```
income_state_mean.head()
```

[208]:
```
                    mean
state
```

```
Alabama     67030.064213
Alaska      92136.545109
Arizona     73328.238798
Arkansas    64765.377850
California  87655.470820
```

[209]:
```python
# calculate rent percentage
rent_percent=rent_state_mean['mean']/income_state_mean['mean']
```

[210]:
```python
rent_percent
```

[210]:
```
state
Alabama                 0.011547
Alaska                  0.012870
Arizona                 0.014970
Arkansas                0.011131
California              0.016783
Colorado                0.013529
Connecticut             0.012637
Delaware                0.012929
District of Columbia    0.013198
Florida                 0.015772
Georgia                 0.013161
Hawaii                  0.018224
Idaho                   0.011957
Illinois                0.012620
Indiana                 0.012022
Iowa                    0.009940
Kansas                  0.011066
Kentucky                0.011068
Louisiana               0.012160
Maine                   0.011674
Maryland                0.013947
Massachusetts           0.012312
Michigan                0.012766
Minnesota               0.011058
Mississippi             0.012428
Missouri                0.011670
Montana                 0.010789
Nebraska                0.010912
Nevada                  0.015242
New Hampshire           0.011949
New Jersey              0.013678
New Mexico              0.012330
New York                0.014410
North Carolina          0.012166
North Dakota            0.009303
```

```
Ohio                      0.011401
Oklahoma                  0.011632
Oregon                    0.013253
Pennsylvania              0.011902
Puerto Rico               0.015133
Rhode Island              0.012292
South Carolina            0.012657
South Dakota              0.009192
Tennessee                 0.012286
Texas                     0.012899
Utah                      0.013192
Vermont                   0.011743
Virginia                  0.014050
Washington                0.013352
West Virginia             0.010341
Wisconsin                 0.011189
Wyoming                   0.010785
Name: mean, dtype: float64
```

Perform correlation analysis for all the relevant variables by creating a heatmap. Describe your findings.

[211]:
```
df_train.columns
```

[211]:
```
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_bin', 'pop_binss'],
      dtype='object')
```

[212]:
```
df_num=df_train.select_dtypes(exclude='object')
```

[213]:
```
df_num.shape
```

[213]: (27321, 75)

[214]:
```
df_num.corr()
```

[214]:

	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	-0.062258	-0.021734	-0.125971	-0.024814	-0.009643	0.102885	

	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	0.031327	0.001078	-0.197203	-0.205738	…	

	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	-0.266785	-0.183109	0.458191	0.388943	

	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     -0.152949  -0.086122  0.206957  0.014572    -0.156631

             age_median
COUNTYID      -0.062258
STATEID       -0.021734
zip_code      -0.125971
area_code     -0.024814
lat           -0.009643
…                  …
separated     -0.086122
divorced       0.206957
bad_debt       0.014572
pop_density   -0.156631
age_median     1.000000

[74 rows x 74 columns]
```

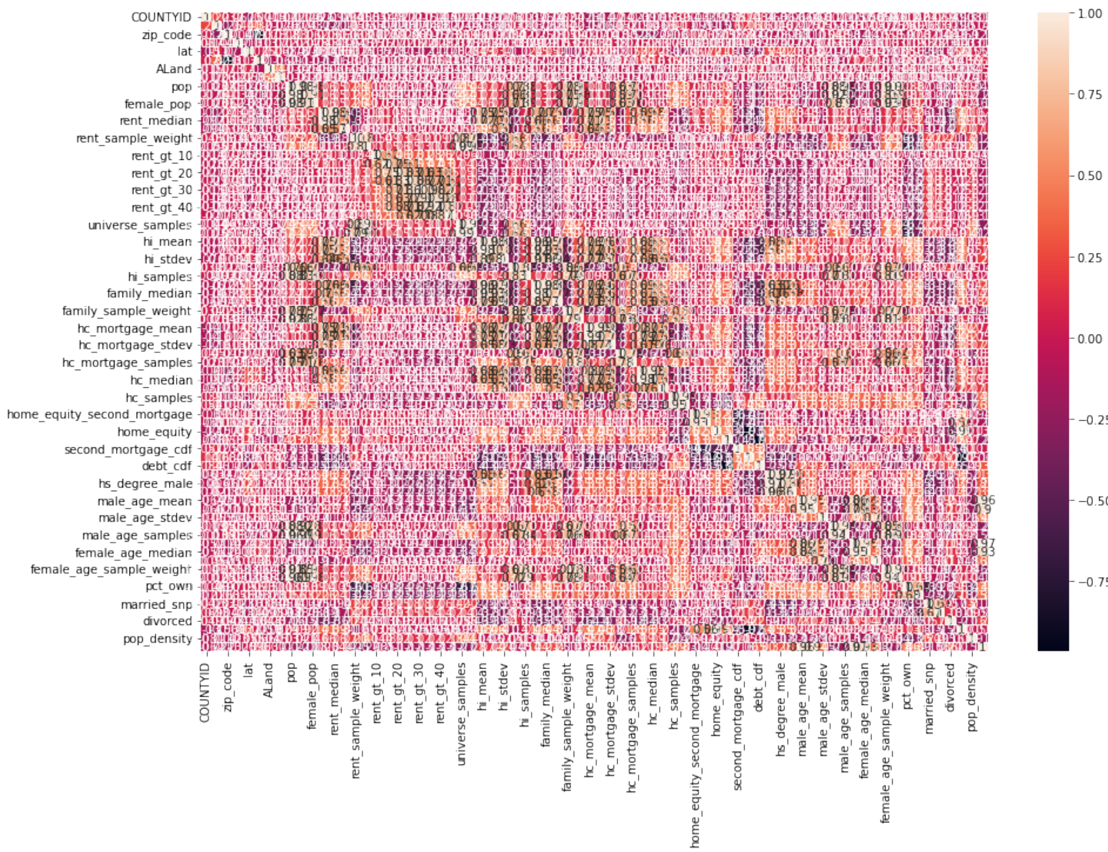
[215]:

```
# cols=df_train[['']].corr()
```

[216]:

```
plt.figure(figsize=(15,10))
sns.heatmap(df_num.corr(),annot=True)
```

[216]: <AxesSubplot:>



[217]:
```
df_train.corr().nlargest(15,'hc_mortgage_mean')
```

[217]:

	COUNTYID	STATEID	zip_code	area_code	lat	\
hc_mortgage_mean	-0.139581	-0.167274	-0.016521	0.042561	0.097747	
hc_mortgage_median	-0.137223	-0.163141	-0.014076	0.040420	0.098932	
hc_mortgage_stdev	-0.121160	-0.161088	-0.017648	0.037865	0.062863	
hc_mean	-0.090427	-0.014471	-0.216220	0.032167	0.217543	
hc_median	-0.090027	-0.006556	-0.218867	0.032809	0.216665	
hi_stdev	-0.076096	-0.102172	-0.008421	0.003285	0.107065	
hi_mean	-0.078694	-0.085679	0.001909	0.018253	0.128503	
family_mean	-0.075688	-0.071612	-0.024658	0.001865	0.151403	
rent_mean	-0.099668	-0.215943	0.073246	0.042648	-0.004272	
family_median	-0.073908	-0.062530	-0.027690	0.002106	0.150768	
hi_median	-0.077105	-0.075635	0.002730	0.022112	0.134177	
rent_median	-0.097069	-0.210061	0.066309	0.042963	-0.006604	
family_stdev	-0.061587	-0.094180	-0.013424	-0.005350	0.111685	
rent_stdev	-0.093584	-0.160124	0.036273	0.005963	0.052528	
hc_stdev	-0.055779	-0.059510	-0.090844	0.026916	0.085814	

```
                         lng     ALand    AWater       pop  male_pop  …  \
```

hc_mortgage_mean	-0.097289	-0.056334	-0.009922	0.110659	0.106709	…
hc_mortgage_median	-0.098047	-0.057950	-0.010905	0.106507	0.102745	…
hc_mortgage_stdev	-0.081923	-0.015402	0.005098	0.082230	0.079537	…
hc_mean	0.151952	-0.056723	-0.010573	0.051515	0.040595	…
hc_median	0.157308	-0.058138	-0.010907	0.050546	0.039426	…
hi_stdev	-0.047004	-0.018233	0.000892	0.126602	0.120234	…
hi_mean	-0.057359	-0.028435	-0.002166	0.166913	0.166467	…
family_mean	-0.027104	-0.027897	-0.002058	0.128173	0.125614	…
rent_mean	-0.168511	-0.067169	-0.009534	0.160590	0.156952	…
family_median	-0.022271	-0.029353	-0.002436	0.124272	0.121873	…
hi_median	-0.056426	-0.029742	-0.002153	0.173015	0.173463	…
rent_median	-0.158885	-0.065507	-0.009345	0.154733	0.152130	…
family_stdev	-0.035724	-0.017816	0.001787	0.106943	0.100588	…
rent_stdev	-0.125884	-0.033488	0.002494	0.116900	0.106605	…
hc_stdev	0.036454	-0.006305	0.004771	0.051414	0.045674	…

	female_age_sample_weight	female_age_samples	pct_own	\
hc_mortgage_mean	0.089454	0.111564	0.067828	
hc_mortgage_median	0.085296	0.107336	0.057242	
hc_mortgage_stdev	0.056719	0.082654	0.150366	
hc_mean	0.041283	0.061084	0.102150	
hc_median	0.041768	0.060374	0.089392	
hi_stdev	0.080518	0.128452	0.380186	
hi_mean	0.099221	0.162200	0.481066	
family_mean	0.081742	0.127229	0.450961	
rent_mean	0.127662	0.159766	0.140249	
family_median	0.078094	0.123292	0.451739	
hi_median	0.102366	0.167360	0.493401	
rent_median	0.119717	0.153037	0.131696	
family_stdev	0.073927	0.110206	0.313391	
rent_stdev	0.108470	0.123760	0.050962	
hc_stdev	0.032852	0.055516	0.109607	

	married	married_snp	separated	divorced	bad_debt	\
hc_mortgage_mean	0.222728	-0.082061	-0.178431	-0.403366	0.472699	
hc_mortgage_median	0.207688	-0.074806	-0.170123	-0.397459	0.462500	
hc_mortgage_stdev	0.273710	-0.112352	-0.180225	-0.296222	0.381657	
hc_mean	0.199810	-0.116247	-0.167693	-0.336902	0.360709	
hc_median	0.185114	-0.110327	-0.160633	-0.328496	0.345310	
hi_stdev	0.444157	-0.253367	-0.282948	-0.343387	0.414195	
hi_mean	0.530892	-0.291916	-0.316511	-0.390061	0.467399	
family_mean	0.480095	-0.314925	-0.323433	-0.353274	0.455988	
rent_mean	0.255671	-0.106256	-0.188108	-0.374508	0.412618	
family_median	0.473053	-0.310826	-0.314345	-0.346997	0.442937	
hi_median	0.533164	-0.291775	-0.310595	-0.385085	0.458846	
rent_median	0.242283	-0.094454	-0.174294	-0.358431	0.390721	
family_stdev	0.368075	-0.238841	-0.261667	-0.287778	0.386193	

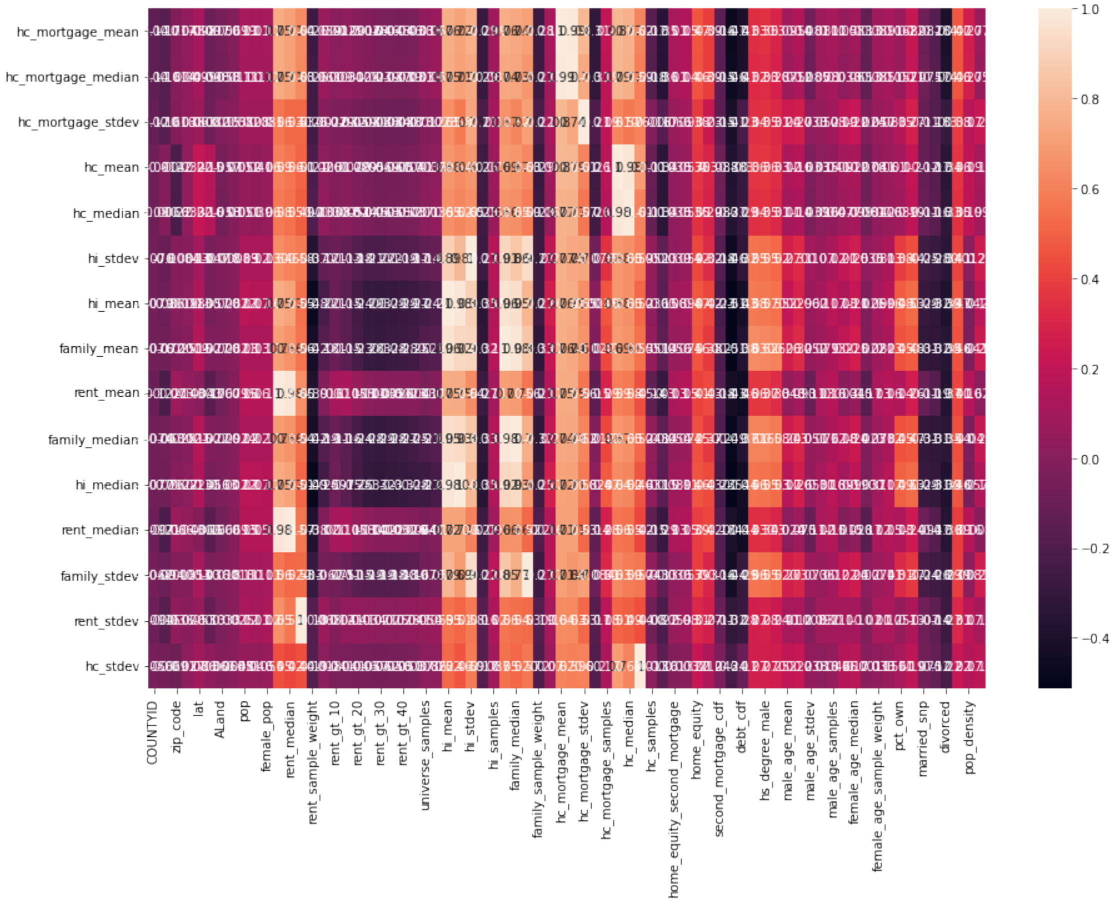
```
rent_stdev           0.131131    -0.069796  -0.136709 -0.267751  0.307788
hc_stdev             0.187453    -0.074519  -0.121528 -0.218639  0.219893

                     pop_density  age_median
hc_mortgage_mean        0.266100    0.075035
hc_mortgage_median      0.269361    0.055619
hc_mortgage_stdev       0.171223    0.224570
hc_mean                 0.190739    0.111940
hc_median               0.188590    0.094406
hi_stdev                0.011956    0.244971
hi_mean                -0.041501    0.174272
family_mean            -0.040661    0.237214
rent_mean               0.156928    0.023491
family_median          -0.040476    0.214170
hi_median              -0.057207    0.139512
rent_median             0.156798    0.004396
family_stdev            0.008050    0.253911
rent_stdev              0.173640    0.110887
hc_stdev                0.165257    0.185324

[15 rows x 74 columns]
```

[218]:

```python
plt.figure(figsize=(15,10))
sns.heatmap(df_train.corr().nlargest(15,'hc_mortgage_mean'),annot=True)
```

[218]:

```
<AxesSubplot:>
```



[222]:
```
df_train.dtypes
```

[222]:
```
COUNTYID            int64
STATEID             int64
state              object
state_ab           object
city               object
                    …
bad_debt          float64
pop_density       float64
age_median        float64
pop_bin            object
pop_binss        category
Length: 82, dtype: object
```

Data Pre-processing:

The economic multivariate data has a significant number of measured variables. The goal is to find where the measured variables depend on a number of smaller unobserved common factors or latent variables.

Each variable is assumed to be dependent upon a linear combination of the common factors, and the coefficients are known as loadings. Each measured variable also includes a component due to independent random variability, known as "specific variance" because it is specific to one variable. Obtain the common factors and then plot the loadings. Use factor analysis to find latent variables in our dataset and gain insight into the linear relationships in the data.

`Following are the list of latent variables:`

Highschool graduation rates

Median population age

Second mortgage statistics

Percent own

Bad debt expense

[219]:
```
pip install factor_analyzer
```

```
Requirement already satisfied: factor_analyzer in
c:\users\shibn\anaconda3\lib\site-packages (0.4.1)
Requirement already satisfied: scikit-learn in
c:\users\shibn\anaconda3\lib\site-packages (from factor_analyzer) (1.0.2)
Requirement already satisfied: scipy in c:\users\shibn\anaconda3\lib\site-
packages (from factor_analyzer) (1.7.3)
Requirement already satisfied: numpy in c:\users\shibn\anaconda3\lib\site-
packages (from factor_analyzer) (1.21.5)
Requirement already satisfied: pre-commit in c:\users\shibn\anaconda3\lib\site-
packages (from factor_analyzer) (3.2.0)
Requirement already satisfied: pandas in c:\users\shibn\anaconda3\lib\site-
packages (from factor_analyzer) (1.4.2)
Requirement already satisfied: pytz>=2020.1 in
c:\users\shibn\anaconda3\lib\site-packages (from pandas->factor_analyzer)
(2021.3)
Requirement already satisfied: python-dateutil>=2.8.1 in
c:\users\shibn\anaconda3\lib\site-packages (from pandas->factor_analyzer)
(2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\shibn\anaconda3\lib\site-
packages (from python-dateutil>=2.8.1->pandas->factor_analyzer) (1.16.0)
Requirement already satisfied: virtualenv>=20.10.0 in
c:\users\shibn\anaconda3\lib\site-packages (from pre-commit->factor_analyzer)
(20.21.0)
Requirement already satisfied: nodeenv>=0.11.1 in
c:\users\shibn\anaconda3\lib\site-packages (from pre-commit->factor_analyzer)
(1.7.0)
Requirement already satisfied: pyyaml>=5.1 in c:\users\shibn\anaconda3\lib\site-
packages (from pre-commit->factor_analyzer) (6.0)
Requirement already satisfied: cfgv>=2.0.0 in c:\users\shibn\anaconda3\lib\site-
packages (from pre-commit->factor_analyzer) (3.3.1)
Requirement already satisfied: identify>=1.0.0 in
```

```
c:\users\shibn\anaconda3\lib\site-packages (from pre-commit->factor_analyzer)
(2.5.21)
Requirement already satisfied: setuptools in c:\users\shibn\anaconda3\lib\site-
packages (from nodeenv>=0.11.1->pre-commit->factor_analyzer) (61.2.0)
Requirement already satisfied: distlib<1,>=0.3.6 in
c:\users\shibn\anaconda3\lib\site-packages (from virtualenv>=20.10.0->pre-
commit->factor_analyzer) (0.3.6)
Requirement already satisfied: filelock<4,>=3.4.1 in
c:\users\shibn\anaconda3\lib\site-packages (from virtualenv>=20.10.0->pre-
commit->factor_analyzer) (3.6.0)
Requirement already satisfied: platformdirs<4,>=2.4 in
c:\users\shibn\anaconda3\lib\site-packages (from virtualenv>=20.10.0->pre-
commit->factor_analyzer) (3.1.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in
c:\users\shibn\anaconda3\lib\site-packages (from scikit-learn->factor_analyzer)
(2.2.0)
Requirement already satisfied: joblib>=0.11 in
c:\users\shibn\anaconda3\lib\site-packages (from scikit-learn->factor_analyzer)
(1.1.0)
Note: you may need to restart the kernel to use updated packages.
```

[220]:
```
from factor_analyzer import FactorAnalyzer
```

[221]:
```
fa=FactorAnalyzer(n_factors=5)
fa.fit_transform(df_train.select_dtypes(exclude=('object','category')))
```

[221]:
```
array([[-0.53225966,  0.4995425 ,  0.06927542, -1.23571292, -0.17578897],
       [-0.77438054, -0.41924946,  2.04708799,  0.3059811 ,  1.41839524],
       [-0.12312115,  1.01842054,  0.09125775, -0.64251645, -0.7597503 ],
       …,
       [ 0.06519723, -0.69914819,  1.40652446, -1.3416408 ,  0.54466455],
       [ 2.55967014,  3.1886579 ,  4.84563844,  0.09284848, -1.13322257],
       [-0.4976945 , -0.2305192 , -3.30946189,  0.01809383, -1.53313422]])
```

Data Modeling :

Build a linear Regression model to predict the total monthly expenditure for home mortgages loan.

```
Please refer deplotment_RE.xlsx. Column hc_mortgage_mean is predicted variable. This is the

Note: Exclude loans from prediction model which have NaN (Not a Number) values for hc_mortga

a) Run a model at a Nation level. If the accuracy levels and R square are not satisfactory p

b) Run another model at State level. There are 52 states in USA.

c) Keep below considerations while building a linear regression model:
```

Variables should have significant impact on predicting Monthly mortgage and owner costs

Utilize all predictor variable to start with initial hypothesis

R square of 60 percent and above should be achieved

Ensure Multi-collinearity does not exist in dependent variables

Test if predicted variable is normally distributed

[223]:
```python
df_train.columns
```

[223]:
```
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_bin', 'pop_binss'],
      dtype='object')
```

[227]:
```python
df_train['type'].value_counts()
```

[227]:
```
City       15237
Town        3666
CDP         3658
Village     3216
Borough     1226
Urban        318
Name: type, dtype: int64
```

[234]:
```python
#convert type column into numerical data
df_train.replace({'City':1,'Town':2,'CDP':3,'Village':4,'Borough':5,'Urban':6},␣
 ↪inplace=True)
```

[235]:
```python
df_train['type'].value_counts()
```

[235]:
```
1    15237
2     3666
```

```
3    3658
4    3216
5    1226
6     318
Name: type, dtype: int64
```

[236]:
```
#convert type column into numerical data
df_test.replace({'City':1,'Town':2,'CDP':3,'Village':4,'Borough':5,'Urban':6},␣
↪inplace=True)
```

[238]:
```
df_test['type'].value_counts()
```

[238]:
```
1    6481
2    1634
3    1558
4    1356
5     509
6     171
Name: type, dtype: int64
```

[239]:
```
df_train.columns
```

[239]:
```
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_bin', 'pop_binss'],
      dtype='object')
```

[242]:
```
input_cols=['COUNTYID',␣
↪'STATEID','type','zip_code','pop','family_mean','second_mortgage',␣
↪'home_equity','debt','hs_degree','age_median','pct_own','married','separated',␣
↪'divorced']
```

[243]:
```
x_train=df_train[input_cols]
```

[244]:
```
x_train
```

[244]:
```
        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   43.486015  0.79046
246444          0.02222      0.04274  0.60855    0.90487   35.665595  0.52483
245683          0.00000      0.09512  0.73484    0.94288   40.769820  0.85331
279653          0.01086      0.01086  0.52714    0.91500   48.211375  0.65037
247218          0.05426      0.05426  0.51938    1.00000   25.126130  0.13046
…                   …          …      …         …          …          …
279212          0.00000      0.00000  0.11694    0.60155   42.435825  0.60422
277856          0.02112      0.19641  0.65364    0.95737   37.983820  0.68072
233000          0.02024      0.07857  0.58095    0.93555   41.706760  0.78508
287425          0.07550      0.12556  0.65722    0.98540   40.006650  0.93970
265371          0.01412      0.18362  0.65537    0.87370   34.631955  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]
```

[245]:
```
y_train=df_train['hc_mortgage_mean']
```

[246]:
```
y_train
```

[246]:
```
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
```

[247]:
```
x_test=df_test[input_cols]
```

[248]:
```
x_test
```

[248]:
```
        COUNTYID  STATEID  type  zip_code   pop   family_mean  \
UID
255504       163       26     3     48239  3417   53802.87122
252676         1       23     1      4210  3796   85642.22095
276314        15       42     5     14871  3944   65694.06582
248614       231       21     1     42633  2508   44156.38709
286865       355       48     2     78410  6230  123527.02420
…            …      …    …         …    …            …
238088       105       12     1     33810  5611   70786.81912
242811        31       17     4     60609  2695   38912.54156
250127         9       25     1      1841  7392   99484.96572
241096        27       19     1     51401  5945   75066.29009
287763       453       48     2     78745  4117   54913.24441

        second_mortgage  home_equity     debt  hs_degree  age_median  pct_own  \
UID
255504          0.06443      0.07651  0.63624    0.91047   34.079065  0.70252
252676          0.01175      0.14375  0.64755    0.94290   44.060655  0.85128
276314          0.01316      0.06497  0.45395    0.89238   40.720435  0.81897
248614          0.00995      0.01741  0.41915    0.60908   43.314190  0.84609
286865          0.00000      0.03440  0.63188    0.86297   41.399595  0.79077
…                     …            …       …          …          …        …
238088          0.03619      0.04044  0.43593    0.92097   52.273950  0.93121
242811          0.05909      0.08182  0.63182    0.54890   33.041570  0.33122
250127          0.02727      0.13545  0.74273    0.94057   39.698240  0.84372
```

```
241096          0.03570      0.07967  0.65546    0.91407    42.406990  0.83330
287763          0.00000      0.05042  0.63866    0.78685    35.781795  0.52587

        married  separated  divorced
UID
255504  0.28217    0.03813   0.14299
252676  0.64221    0.00000   0.13377
276314  0.59961    0.01358   0.10026
248614  0.56953    0.04694   0.12489
286865  0.57620    0.00588   0.16379
…          …          …        …
238088  0.65969    0.02135   0.08780
242811  0.42882    0.02829   0.05305
250127  0.50269    0.00108   0.07294
241096  0.66699    0.00000   0.04694
287763  0.51922    0.02520   0.10586

[11709 rows x 15 columns]
```

[249]:
```
y_test=df_test['hc_mortgage_mean']
```

[250]:
```
y_test
```

[250]:
```
UID
255504    1139.24548
252676    1533.25988
276314    1254.54462
248614     862.65763
286865    1996.41425
             …
238088    1269.83033
242811    1406.83478
250127    1791.63902
241096    1182.30365
287763    1364.17379
Name: hc_mortgage_mean, Length: 11709, dtype: float64
```

[252]:
```
from sklearn.preprocessing import StandardScaler
sc=StandardScaler()
```

[257]:
```
x_train_scaled=sc.fit_transform(x_train)
```

[258]:
```
x_test_scaled=sc.fit_transform(x_test)
```

[259]:
```
#apply linear regression model
from sklearn.linear_model import LinearRegression
linear_reg=LinearRegression()
```

[260]:
```
linear_reg.fit(x_train_scaled,y_train)
```

[260]:
```
LinearRegression()
```

[261]:
```
y_pred=linear_reg.predict(x_test_scaled)
```

[262]:
```
y_pred
```

[262]:
```
array([ 891.92138104, 1600.33363956, 1074.51799452, …, 1923.90730368,
        1498.7391998 , 1142.69088564])
```

[263]:
```
from sklearn.metrics import mean_squared_error,r2_score,accuracy_score
```

[278]:
```
print('Mean squared error',np.sqrt(mean_squared_error(y_test,y_pred)))
print('R2 score',r2_score(y_test,y_pred))
```

```
Mean squared error 326.9453339479488
R2 score 0.7284747034569053
```

) Run another model at State level. There are 52 states in USA.

```
c) Keep below considerations while building a linear regression model:
```

Variables should have significant impact on predicting Monthly mortgage and owner costs

Utilize all predictor variable to start with initial hypothesis

R square of 60 percent and above should be achieved

Ensure Multi-collinearity does not exist in dependent variables

Test if predicted variable is normally distributed

[266]:
```
df_train['STATEID'].unique()
```

[266]:
```
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], dtype=int64)
```

[267]:
```
for i in [20,36,45]:
    print('state id -->')

    x_train_nation=df_train[df_train['COUNTYID']==i][input_cols]
    y_train_nation=df_train[df_train['COUNTYID']==i]['hc_mortgage_mean']

    x_test_nation=df_test[df_test['COUNTYID']==i][input_cols]
    y_test_nation=df_test[df_test['COUNTYID']==i]['hc_mortgage_mean']
```

```
state id -->
state id -->
```

state id -->

[268]:
```
x_train_scaled_nation=sc.fit_transform(x_train_nation)
```

[269]:
```
x_test_scaled_nation=sc.fit_transform(x_test_nation)
```

[270]:
```
linear_reg.fit(x_train_scaled_nation,y_train_nation)
```

[270]:
```
LinearRegression()
```

[272]:
```
yprd=linear_reg.predict(x_test_scaled_nation)
```

[273]:
```
yprd
```

[273]:
```
array([2524.53624295, 1220.71648113, 1412.90827791, 2452.23843011,
       1042.21928005,  772.71707396, 2813.65559952, 1535.39734496,
       1423.52772043, 1050.1930857 , 2796.2432792 , 1625.41452056,
        931.00673672, 1935.08684662, 1613.69954268, 1465.33462307,
       1480.72223608, 1366.57761683, 1004.95534994, 1598.17459461,
       1293.17592552, 1705.57976051, 1665.93368848, 1461.392865  ,
       1907.24979324, 1102.3769964 , 1179.71295203, 1891.0204914 ,
       1594.24158317, 1628.59440397, 1725.68096021,  964.6349026 ,
       1392.26290179, 1036.48274967, 1118.65256187, 1120.62146281,
       1252.67632674, 1485.2284054 , 1744.28253571, 1020.8495702 ,
       1139.82182776,  833.72319379,  987.24742205, 1984.88646734,
       1395.62060177, 1130.25501782, 1502.08126331, 1310.43541018,
       1081.07454977,  876.31347685, 1330.44130526, 1589.80799971,
       1246.8362186 , 1214.57310709, 1339.05217266, 1152.64384833,
       1402.70001576, 1128.6496991 , 1191.40447023, 2252.44878616,
       1095.25321235, 1647.27378731, 1436.17091933, 1295.79430389,
       1190.15588539, 1389.63350028, 1302.2477948 , 1599.14508562,
       1218.04561306,  980.81534656, 1357.9687197 , 1445.07631287,
       1046.08009761, 1014.49799237, 1520.05023673, 1765.53199504,
       2032.63626808, 1781.69954681, 1978.04197348,  907.98335223,
       1222.09762746, 1655.52852385, 2087.70309673, 1860.68974841,
       1919.35791637, 1234.52855122, 1500.3575633 , 1099.01535697,
       1044.09529234, 1311.98645144, 2470.07488251, 1345.89646308,
       1713.62105176, 1874.9023845 ,  769.53128322, 1058.64336727,
       1293.27183427, 1070.6295533 , 1380.55548671, 2255.94107541,
       1523.51479707, 1133.46482792, 2750.16454955, 1305.02277854,
       1283.86626724, 1401.8356114 , 1249.982845  , 1276.33667174,
       1218.85559381, 1592.92521027, 1594.42063997, 1566.23179984,
       1266.01147813, 1663.88925981, 2108.31801414, 1450.41571971])
```

[274]:
```
print('Mean squared error',np.sqrt(mean_squared_error(y_test_nation,yprd)))
```

```
Mean squared error 218.51484128047764
```

[275]:
```python
print('R2 score',r2_score(y_test_nation,yprd))
```

```
R2 score 0.8019744277288993
```

[ ]:
```python

```