real-estate

May 1, 2023

[1]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

[2]:
```python
#loading train and test data
train_df = pd.read_csv(r'C:\Users\nilesh\Downloads\Project_1\train.csv')
test_df = pd.read_csv(r'C:\Users\nilesh\Downloads\Project_1\test.csv')
```

[3]:
```python
train_df.head()
```

[3]:
```
      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]
```

[4]:
```
test_df.head()
```

[4]:
```
      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]
```

[5]:
```
train_df.shape
```

[5]:
```
(27321, 80)
```

[6]:
```
test_df.shape
```

[6]:
```
(11709, 80)
```

[7]:
```
train_df.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
```

[8]:
```
train_df.columns
```

[8]:
```
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')
```

[9]:
```
#This flag will help us split the data back later
train_df['split']= 'Train'
test_df['split']= 'Test'
```

[10]:
```
df_combined=train_df.append(test_df, ignore_index=True)
df_combined.head()
```

```
C:\Users\nilesh\AppData\Local\Temp\ipykernel_8940\3473088661.py:1:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  df_combined=train_df.append(test_df, ignore_index=True)
```

[10]:
```
      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_median  female_age_stdev  \
0    Hamilton        Hamilton   City  …          45.33333          22.51276
1  South Bend        Roseland   City  …          37.58333          23.43353
2    Danville        Danville   City  …          42.83333          23.94119
3    San Juan        Guaynabo  Urban  …          50.58333          24.32015
4   Manhattan  Manhattan City   City  …          21.58333          11.10484

   female_age_sample_weight  female_age_samples  pct_own  married  \
0                 685.33845              2618.0  0.79046  0.57851
1                 267.23367              1284.0  0.52483  0.34886
2                 707.01963              3238.0  0.85331  0.64745
3                 362.20193              1559.0  0.65037  0.47257
4                1854.48652              3051.0  0.13046  0.12356
```

```
   married_snp  separated  divorced  split
0      0.01882    0.01240   0.08770  Train
1      0.01426    0.01426   0.09030  Train
2      0.02830    0.01607   0.10657  Train
3      0.02021    0.02021   0.10106  Train
4      0.00000    0.00000   0.03109  Train

[5 rows x 81 columns]
```

[11]:
```
df_combined.tail()
```

[11]:
```
          UID  BLOCKID  SUMLEVEL  COUNTYID  STATEID          state state_ab  \
39025  238088      NaN       140       105       12        Florida       FL
39026  242811      NaN       140        31       17       Illinois       IL
39027  250127      NaN       140         9       25  Massachusetts       MA
39028  241096      NaN       140        27       19           Iowa       IA
39029  287763      NaN       140       453       48          Texas       TX

           city              place     type  … female_age_median  \
39025  Lakeland    Crystal Springs     City  …          59.58333
39026   Chicago       Chicago City  Village  …          32.83333
39027  Lawrence  Methuen Town City     City  …          43.66667
39028   Carroll       Carroll City     City  …          48.16667
39029    Austin  Sunset Valley City    Town  …          35.41667

       female_age_stdev  female_age_sample_weight  female_age_samples  \
39025          23.23426                 699.33353              2914.0
39026          20.24698                 306.63915              1191.0
39027          23.17995                 900.13903              3723.0
39028          24.84209                 693.82905              3213.0
39029          20.68049                 559.30291              2047.0

       pct_own  married  married_snp  separated  divorced  split
39025  0.93121  0.65969      0.02135    0.02135   0.08780   Test
39026  0.33122  0.42882      0.07781    0.02829   0.05305   Test
39027  0.84372  0.50269      0.00108    0.00108   0.07294   Test
39028  0.83330  0.66699      0.02738    0.00000   0.04694   Test
39029  0.52587  0.51922      0.08066    0.02520   0.10586   Test

[5 rows x 81 columns]
```

[12]:
```
df_combined.shape
```

[12]:
```
(39030, 81)
```

[13]:
```
df_combined.isna().sum()
```

[13]:
```
UID                 0
BLOCKID         39030
SUMLEVEL            0
COUNTYID            0
STATEID             0
                ...
married           275
married_snp       275
separated         275
divorced          275
split               0
Length: 81, dtype: int64
```

[14]:
```
# Fill rate of the variables -> (1- missing %)
1-df_combined.isna().sum()/len(df_combined)
```

[14]:
```
UID            1.000000
BLOCKID        0.000000
SUMLEVEL       1.000000
COUNTYID       1.000000
STATEID        1.000000
                 ...
married        0.992954
married_snp    0.992954
separated      0.992954
divorced       0.992954
split          1.000000
Length: 81, dtype: float64
```

[15]:
```
# BlOCKID is completly missing or null in both train and test data. So we will␣
↪drop BLOCKID feature.
df_combined.drop(columns =['BLOCKID'], inplace=True)
```

[16]:
```
df_combined.isna().sum()/len(df_combined)*100
```

[16]:
```
UID            0.000000
SUMLEVEL       0.000000
COUNTYID       0.000000
STATEID        0.000000
state          0.000000
                 ...
married        0.704586
married_snp    0.704586
separated      0.704586
divorced       0.704586
split          0.000000
Length: 80, dtype: float64
```

[17]:
```python
# Missing value greater than zero
col_check=df_combined.isna().sum().to_frame().reset_index()
null_col=col_check[col_check[0]>0]['index'].tolist()
null_col
```

[17]:
```
['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']
```

[18]:

```
#If the feature have less than 8 unique value then I am consdering as␣
 ↪categorical else it will be continuous
for i in null_col:
    print(i)
    if df_combined[i].nunique()>8:      #Continuous data
        df_combined[i].fillna(df_combined[i].median(),inplace=True)     #Bcz␣
 ↪median is not impacted by outlier
    else:df_combined[i].fillna(df_combined[i].mode()[0],inplace=True) ␣
 ↪#Categorical data
```

```
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
```

[19]:
```
df_combined.isna().sum()/len(df_combined)*100
```

[19]:
```
UID            0.0
SUMLEVEL       0.0
COUNTYID       0.0
STATEID        0.0
state          0.0
              …
```

```
married        0.0
married_snp    0.0
separated      0.0
divorced       0.0
split          0.0
Length: 80, dtype: float64
```

[20]:
```
# Drop duplicate observations
df_combined.drop_duplicates(inplace=True)
df_combined.shape
```

[20]:
```
(38838, 80)
```

[21]:
```
# As we have seen above we have 123 unique UID which are common in both train␣
↪and test data. so duplicate UID removing them.
df_combined.drop_duplicates(subset=['UID'],inplace=True)
df_combined.shape
```

[21]:
```
(38715, 80)
```

0.1 4. EDA

a. Explore the top 2,500 locations where the percentage of households with a second mortgage is the highest and percent ownership is above 10%.

[22]:
```
top_location = train_df[(train_df['second_mortgage']<0.5) &␣
↪(train_df['pct_own']> 0.1)].
↪sort_values(by='second_mortgage',ascending=False).head(2500)
top_location
```

[22]:
```
          UID  BLOCKID  SUMLEVEL  COUNTYID  STATEID           state state_ab  \
11980  251185      NaN       140        27       25   Massachusetts       MA
26018  269323      NaN       140        81       36        New York       NY
7829   251324      NaN       140         3       24        Maryland       MD
2077   235788      NaN       140        57       12         Florida       FL
1701   242304      NaN       140        31       17        Illinois       IL
…         …      …        …        …        …              …       …
17914  261444      NaN       140       183       37  North Carolina       NC
5478   225977      NaN       140        37        6      California       CA
25642  251433      NaN       140         5       24        Maryland       MD
26671  278341      NaN       140       101       42    Pennsylvania       PA
24443  230480      NaN       140        77        6      California       CA

              city             place  type  … female_age_median  \
11980    Worcester    Worcester City  City  …          26.16667
26018       Corona      Harbor Hills  City  …          27.66667
7829   Glen Burnie       Glen Burnie   CDP  …          30.66667
2077         Tampa  Egypt Lake-leto  City  …          28.58333
```

```
1701          Chicago        Lincolnwood  Village  …         39.83333
…                …                 …        …  …           …
17914         Raleigh       Raleigh City  Village  …         25.00000
5478   Marina Del Rey     Marina Del Rey     City  …         41.41667
25642       Baltimore           Lochearn      CDP  …         52.75000
26671    Philadelphia  Philadelphia City  Borough  …         28.41667
24443         Manteca       Manteca City     City  …         38.83333

       female_age_stdev  female_age_sample_weight  female_age_samples  \
11980          19.21553                 262.09529               994.0
26018          18.45616                 448.69061              1932.0
7829           19.61959                 694.10357              2881.0
2077           18.56943                 814.45000              2684.0
1701           21.71686                 374.52605              1802.0
…                   …                       …                  …
17914          13.44444                1044.70191              2965.0
5478           18.58900                 343.62694              1590.0
25642          24.90042                 301.08168              1323.0
26671          20.68431                 898.30792              3673.0
24443          22.82683                 744.85694              3095.0

       pct_own  married  married_snp  separated  divorced  split
11980  0.20247  0.37844      0.11976    0.09341   0.10539  Train
26018  0.15618  0.44490      0.14555    0.02357   0.04066  Train
7829   0.22380  0.58250      0.08321    0.00000   0.01778  Train
2077   0.11618  0.36953      0.12876    0.09957   0.07339  Train
1701   0.14228  0.41366      0.13852    0.01771   0.09677  Train
…         …       …         …       …        …    …
17914  0.12827  0.23974      0.07685    0.00827   0.07165  Train
5478   0.44682  0.27404      0.04473    0.02057   0.13162  Train
25642  0.84707  0.43002      0.02822    0.00000   0.07223  Train
26671  0.70507  0.28105      0.07121    0.03887   0.09254  Train
24443  0.67116  0.62787      0.06491    0.01817   0.04890  Train

[2500 rows x 81 columns]
```

[23]:

```
top_location=top_location[['COUNTYID', 'STATEID', 'state', 'state_ab', 'city',␣
 ↪'place']]
top_location
```

[23]:

```
       COUNTYID  STATEID          state state_ab          city  \
11980        27       25  Massachusetts       MA     Worcester
26018        81       36       New York       NY        Corona
7829          3       24       Maryland       MD   Glen Burnie
2077         57       12        Florida       FL         Tampa
1701         31       17       Illinois       IL       Chicago
…          …      …             …      …           …
```

```
17914        183        37  North Carolina        NC          Raleigh
5478          37         6      California        CA  Marina Del Rey
25642          5        24        Maryland        MD        Baltimore
26671        101        42    Pennsylvania        PA     Philadelphia
24443         77         6      California        CA          Manteca

                    place
11980      Worcester City
26018        Harbor Hills
7829          Glen Burnie
2077      Egypt Lake-leto
1701          Lincolnwood
…                       …
17914        Raleigh City
5478       Marina Del Rey
25642            Lochearn
26671   Philadelphia City
24443        Manteca City

[2500 rows x 6 columns]
```

b. Bad debt is the debt you should avoid at all costs such as a second mortgage or home equity loan. Conversely, Good debt is all other debt not including second mortgage or home equity loan.

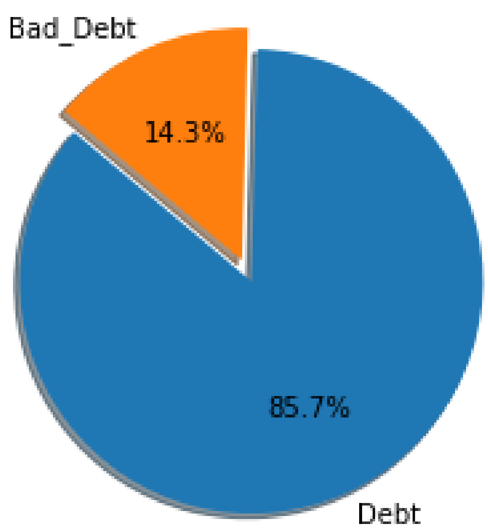
[24]:
```python
df_combined['bad_debt'] = df_combined['second_mortgage'] + df_combined['home_equity'] - df_combined['home_equity_second_mortgage']
df_combined.head()
```

[24]:
```
      UID  SUMLEVEL  COUNTYID  STATEID        state state_ab        city  \
0  267822       140        53       36     New York       NY    Hamilton
1  246444       140       141       18      Indiana       IN  South Bend
2  245683       140        63       18      Indiana       IN    Danville
3  279653       140       127       72  Puerto Rico       PR    San Juan
4  247218       140       161       20       Kansas       KS   Manhattan

           place   type primary  …  female_age_stdev  \
0       Hamilton   City   tract  …          22.51276
1       Roseland   City   tract  …          23.43353
2       Danville   City   tract  …          23.94119
3       Guaynabo  Urban   tract  …          24.32015
4  Manhattan City   City   tract  …          11.10484

   female_age_sample_weight  female_age_samples  pct_own  married  \
0                 685.33845              2618.0  0.79046  0.57851
1                 267.23367              1284.0  0.52483  0.34886
2                 707.01963              3238.0  0.85331  0.64745
```

```
3                  362.20193              1559.0  0.65037  0.47257
4                 1854.48652              3051.0  0.13046  0.12356

   married_snp  separated  divorced  split  bad_debt
0      0.01882    0.01240   0.08770  Train   0.09408
1      0.01426    0.01426   0.09030  Train   0.04274
2      0.02830    0.01607   0.10657  Train   0.09512
3      0.02021    0.02021   0.10106  Train   0.01086
4      0.00000    0.00000   0.03109  Train   0.05426

[5 rows x 81 columns]
```

c. Create pie charts (Venn diagram) to show overall debt (% bad and good debt) and bad debt (2 mortgage and home equity loan)

[25]:
```python
import matplotlib.pyplot as plt
labels = 'Debt', 'Bad_Debt'
x = [df_combined['debt'].mean()*100, df_combined['bad_debt'].mean()*100]
explode = (0.1, 0)  # explode 1st slice

#Plot
plt.pie(x,explode=explode,labels=labels,
autopct='%1.1f%%', shadow=True, startangle=140)

plt.show()
```

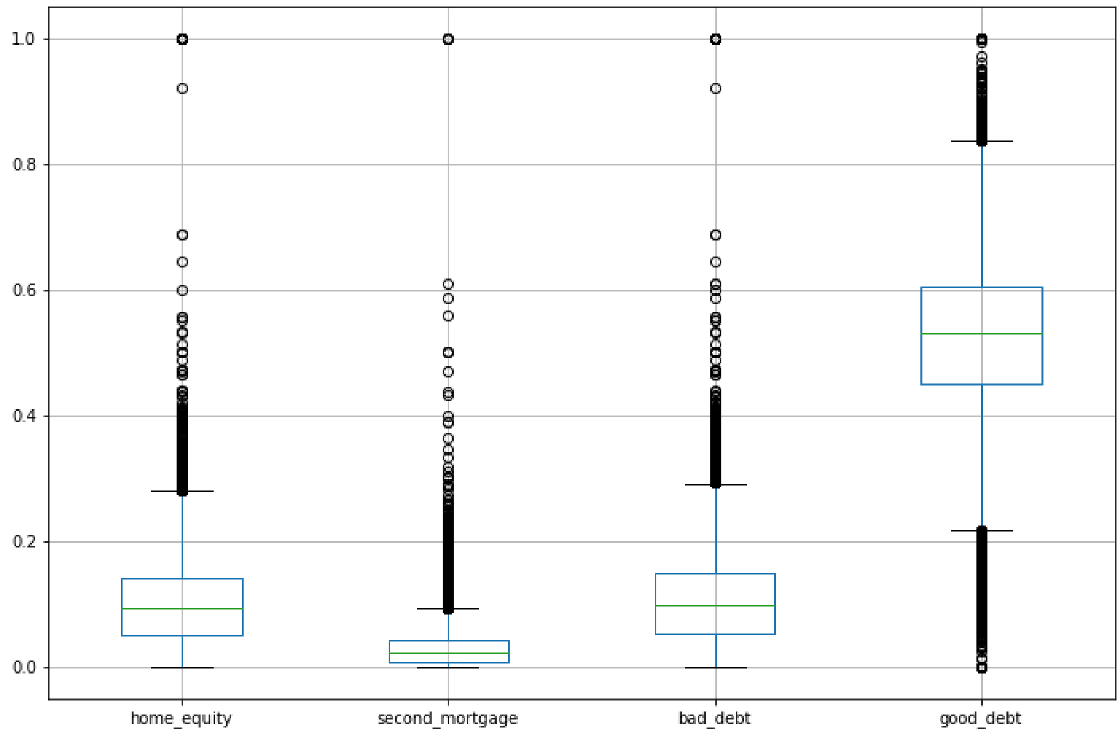


d. Create Box and whisker plot and analyze the distribution for 2nd mortgage, home equity, good debt and bad debt for different cities

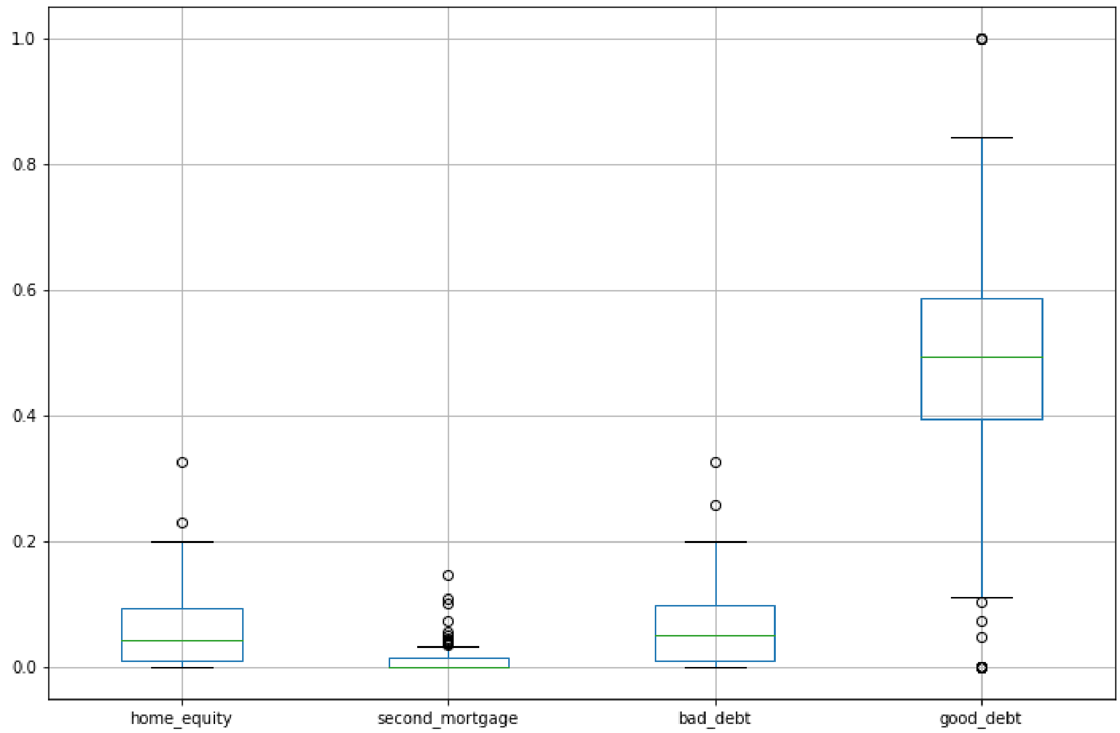
[26]:
```
df_combined['good_debt']=df_combined['debt']-df_combined['bad_debt']
df_combined.head()
```

[26]:
```
      UID  SUMLEVEL  COUNTYID  STATEID        state state_ab        city  \
0  267822       140        53       36     New York       NY    Hamilton
1  246444       140       141       18      Indiana       IN  South Bend
2  245683       140        63       18      Indiana       IN    Danville
3  279653       140       127       72  Puerto Rico       PR    San Juan
4  247218       140       161       20       Kansas       KS   Manhattan

            place   type primary  …  female_age_sample_weight  \
0        Hamilton   City   tract  …                 685.33845
1        Roseland   City   tract  …                 267.23367
2        Danville   City   tract  …                 707.01963
3        Guaynabo  Urban   tract  …                 362.20193
4  Manhattan City   City   tract  …                1854.48652

   female_age_samples  pct_own  married  married_snp  separated  divorced  \
0              2618.0  0.79046  0.57851      0.01882    0.01240   0.08770
1              1284.0  0.52483  0.34886      0.01426    0.01426   0.09030
2              3238.0  0.85331  0.64745      0.02830    0.01607   0.10657
3              1559.0  0.65037  0.47257      0.02021    0.02021   0.10106
4              3051.0  0.13046  0.12356      0.00000    0.00000   0.03109

   split  bad_debt  good_debt
0  Train   0.09408    0.43555
1  Train   0.04274    0.56581
2  Train   0.09512    0.63972
3  Train   0.01086    0.51628
4  Train   0.05426    0.46512

[5 rows x 82 columns]
```

[27]:
```
cities_dist = df_combined[['home_equity','second_mortgage','bad_debt',␣
 ↪'good_debt']]
cities_dist.boxplot( figsize=(12,8),manage_ticks=True, autorange=False, )
plt.show()
```



[28]:
```
new_york=df_combined[df_combined['city']=='New York']
new_york = new_york[['home_equity','second_mortgage','bad_debt', 'good_debt']]
new_york.boxplot( figsize=(12,8),manage_ticks=True, autorange=False, )
plt.show()
```

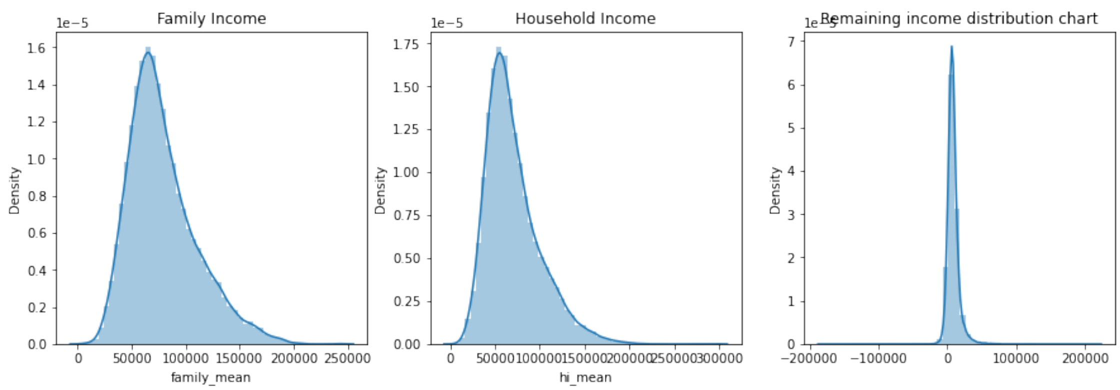


e. Create a collated income distribution chart for family income, house hold income and remaining income

[29]:
```
plt.figure(figsize=(15,10))

plt.subplot(2,3,1)
sns.distplot(train_df['family_mean'])
plt.title('Family Income')
plt.subplot(2,3,2)
sns.distplot(train_df['hi_mean'])
plt.title('Household Income')
plt.subplot(2,3,3)
sns.distplot(train_df['family_mean']-train_df['hi_mean'])
plt.title('Remaining income distribution chart')
plt.show()
```

```
C:\Users\nilesh\anaconda3\lib\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)
C:\Users\nilesh\anaconda3\lib\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)
C:\Users\nilesh\anaconda3\lib\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)
```



6. Population density (hint-use ‘pop’ and ‘Aland’ to calculate)

[30]:
```python
df_combined['population_density']=df_combined['pop']/df_combined['ALand']
df_combined.head()
```

[30]:

	UID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	\
0	267822	140	53	36	New York	NY	Hamilton	
1	246444	140	141	18	Indiana	IN	South Bend	
2	245683	140	63	18	Indiana	IN	Danville	
3	279653	140	127	72	Puerto Rico	PR	San Juan	
4	247218	140	161	20	Kansas	KS	Manhattan	

	place	type	primary	…	female_age_samples	pct_own	married	\
0	Hamilton	City	tract	…	2618.0	0.79046	0.57851	
1	Roseland	City	tract	…	1284.0	0.52483	0.34886	
2	Danville	City	tract	…	3238.0	0.85331	0.64745	
3	Guaynabo	Urban	tract	…	1559.0	0.65037	0.47257	
4	Manhattan City	City	tract	…	3051.0	0.13046	0.12356	

	married_snp	separated	divorced	split	bad_debt	good_debt	\
0	0.01882	0.01240	0.08770	Train	0.09408	0.43555	
1	0.01426	0.01426	0.09030	Train	0.04274	0.56581	
2	0.02830	0.01607	0.10657	Train	0.09512	0.63972	

```
3      0.02021    0.02021   0.10106  Train   0.01086    0.51628
4      0.00000    0.00000   0.03109  Train   0.05426    0.46512

   population_density
0            0.000026
1            0.001687
2            0.000099
3            0.002442
4            0.002207

[5 rows x 83 columns]
```

[31]:

```
df_combined['median_age']=((df_combined['male_age_median'] *␣
↪df_combined['male_pop'])+␣
↪(df_combined['female_age_median']*df_combined['female_pop']))/
↪(df_combined['male_pop']+␣
↪          df_combined['female_pop'])
df_combined.head()
```

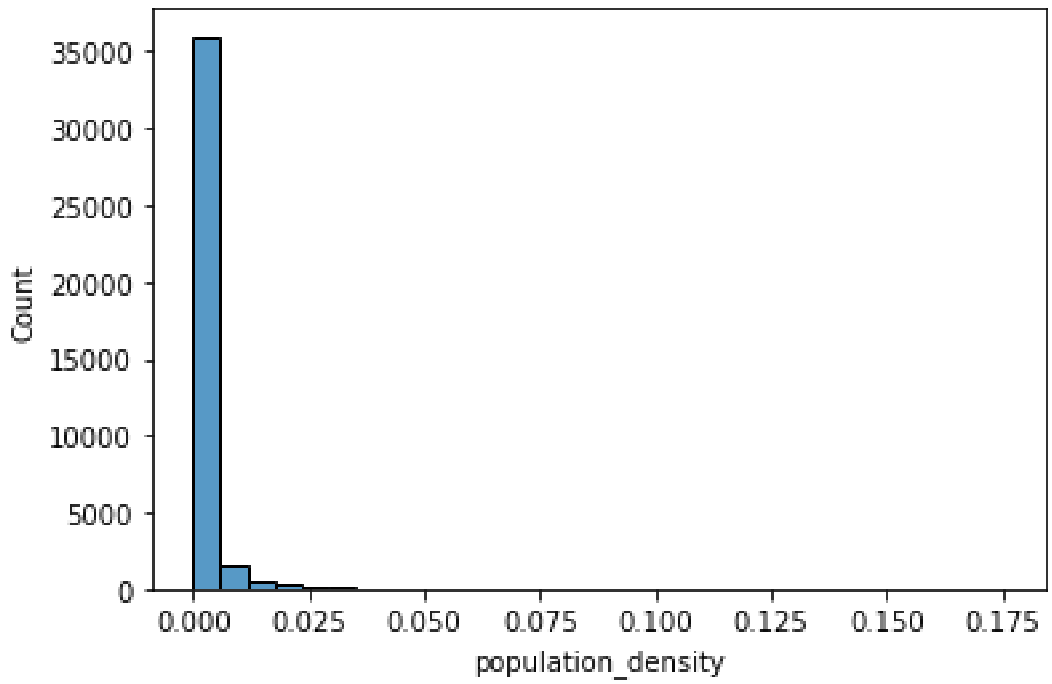
[31]:

```
      UID  SUMLEVEL  COUNTYID  STATEID        state state_ab        city  \
0  267822       140        53       36     New York       NY    Hamilton
1  246444       140       141       18      Indiana       IN  South Bend
2  245683       140        63       18      Indiana       IN    Danville
3  279653       140       127       72  Puerto Rico       PR    San Juan
4  247218       140       161       20       Kansas       KS   Manhattan

            place   type primary  …  pct_own  married  married_snp  \
0        Hamilton   City   tract  …  0.79046  0.57851      0.01882
1        Roseland   City   tract  …  0.52483  0.34886      0.01426
2        Danville   City   tract  …  0.85331  0.64745      0.02830
3        Guaynabo  Urban   tract  …  0.65037  0.47257      0.02021
4  Manhattan City   City   tract  …  0.13046  0.12356      0.00000

   separated  divorced  split  bad_debt  good_debt  population_density  \
0    0.01240   0.08770  Train   0.09408    0.43555            0.000026
1    0.01426   0.09030  Train   0.04274    0.56581            0.001687
2    0.01607   0.10657  Train   0.09512    0.63972            0.000099
3    0.02021   0.10106  Train   0.01086    0.51628            0.002442
4    0.00000   0.03109  Train   0.05426    0.46512            0.002207

   median_age
0   44.667430
1   34.722748
2   41.774472
3   49.879012
```

```
4   21.965629

[5 rows x 84 columns]
```

[32]:

```
sns.histplot(df_combined['population_density'],bins=30)
plt.show()
```

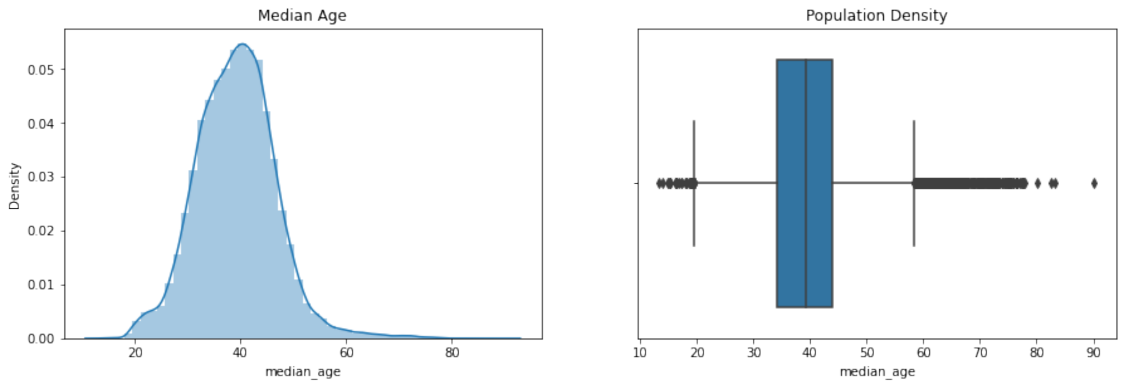


[33]:

```
plt.figure(figsize=(15,10))
plt.subplot(2,2,1)
sns.distplot(df_combined['median_age'])
plt.title('Median Age')
plt.subplot(2,2,2)
sns.boxplot(df_combined['median_age'])
plt.title('Population Density')
plt.show()
```

```
C:\Users\nilesh\anaconda3\lib\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)
C:\Users\nilesh\anaconda3\lib\site-packages\seaborn\_decorators.py:36:
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.
  warnings.warn(
```



7. Create bins for population into a new variable by selecting appropriate class interval so that the no of categories(bins) don't exceed 5 for the ease of analysis.

[34]:
```python
df_combined['pop_bins']=pd.cut(df_combined['pop'],bins=5,labels=['very␣
↪low','low','medium','high','very high'])
df_combined['pop_bins'].value_counts()
```

[34]:
```
very low     38350
low            348
medium          12
high             4
very high        1
Name: pop_bins, dtype: int64
```

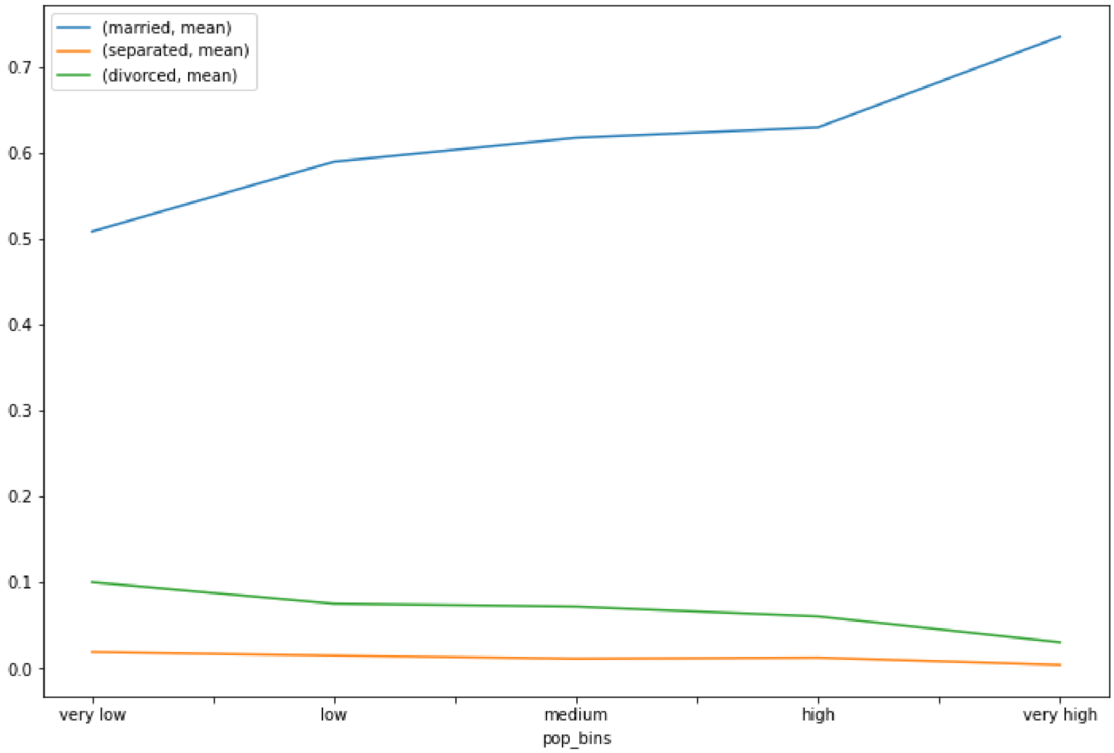
[35]:
```python
df_combined.groupby(by='pop_bins')[['married','separated','divorced']].count()
```

[35]:

pop_bins	married	separated	divorced
very low	38350	38350	38350
low	348	348	348
medium	12	12	12
high	4	4	4
very high	1	1	1

[36]:
```python
plt.figure(figsize=(12,8))
pop_bin_married=df_combined.
↪groupby(by='pop_bins')[['married','separated','divorced']].agg(["mean"])
pop_bin_married.plot(figsize=(12,8))
plt.legend(loc='best')
plt.show()
```

<Figure size 864x576 with 0 Axes>



8.Please detail your observations for rent as a percentage of income at an overall level and for different states.

[37]:
```
rent_mean_state = df_combined.groupby(by='state')['rent_mean'].agg(["mean"])
rent_mean_state.head()
```

[37]:

	mean
state	
Alabama	765.872557
Alaska	1190.093590
Arizona	1084.510940
Arkansas	716.544987
California	1466.020465

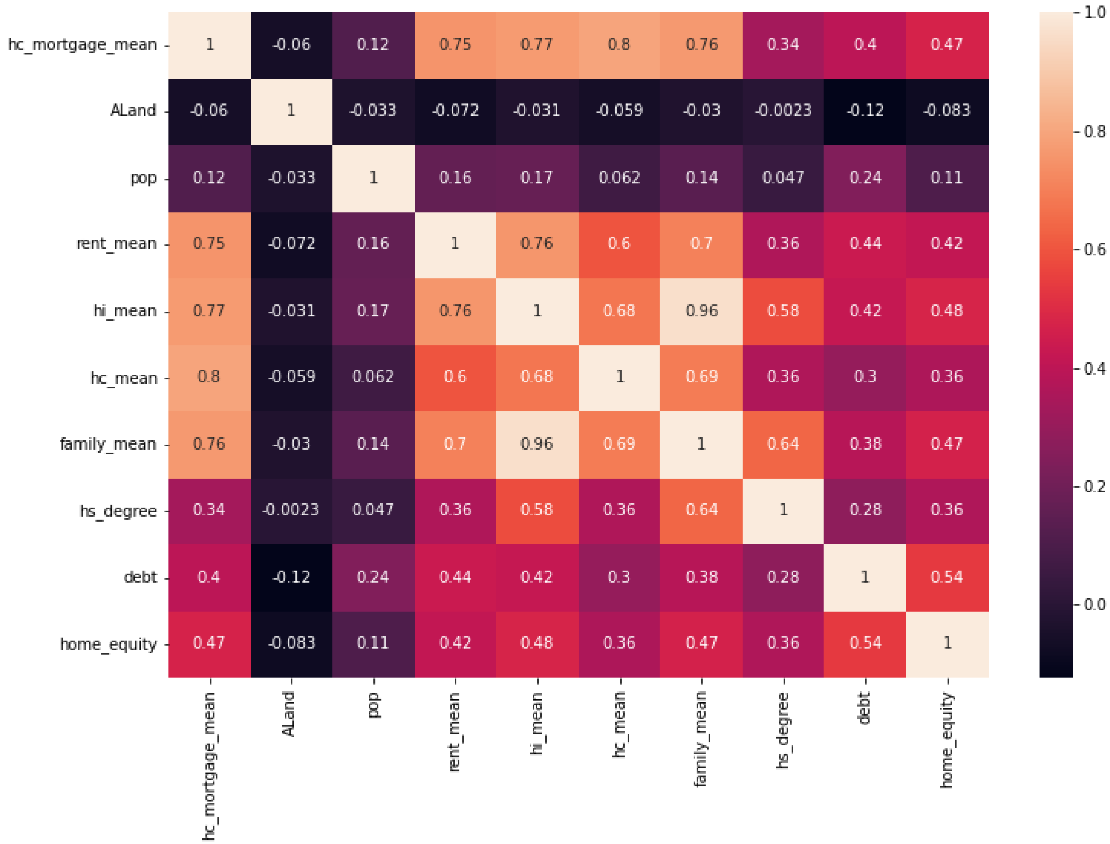
[38]:
```
sum(df_combined['rent_mean'])/sum(train_df['family_mean'])
```

[38]: nan

[39]:
```
train_df.columns
```

[39]:
```
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', 'split'],
      dtype='object')
```

[40]:
```
plt.figure(figsize=(12,8))
sns.
 ↪heatmap(data=df_combined[['hc_mortgage_mean','ALand','pop','rent_mean','hi_mean','hc_mean',
                              'hs_degree','debt','home_equity']].corr(),annot=True)
plt.show()
```



[41]:
```python
train = df_combined[df_combined['split'] == 'Train']
test = df_combined[df_combined['split'] == 'Test']
```

[42]:
```python
train.head()
```

[42]:
```
      UID  SUMLEVEL  COUNTYID  STATEID        state state_ab        city  \
0  267822       140        53       36     New York       NY    Hamilton
1  246444       140       141       18      Indiana       IN  South Bend
2  245683       140        63       18      Indiana       IN    Danville
3  279653       140       127       72  Puerto Rico       PR    San Juan
4  247218       140       161       20       Kansas       KS   Manhattan

            place   type primary  …  married  married_snp  separated  \
0        Hamilton   City   tract  …  0.57851      0.01882    0.01240
1        Roseland   City   tract  …  0.34886      0.01426    0.01426
2        Danville   City   tract  …  0.64745      0.02830    0.01607
3        Guaynabo  Urban   tract  …  0.47257      0.02021    0.02021
4  Manhattan City   City   tract  …  0.12356      0.00000    0.00000

   divorced  split  bad_debt  good_debt  population_density  median_age  \
```

```
0    0.08770  Train   0.09408    0.43555            0.000026    44.667430
1    0.09030  Train   0.04274    0.56581            0.001687    34.722748
2    0.10657  Train   0.09512    0.63972            0.000099    41.774472
3    0.10106  Train   0.01086    0.51628            0.002442    49.879012
4    0.03109  Train   0.05426    0.46512            0.002207    21.965629

   pop_bins
0  very low
1  very low
2  very low
3  very low
4  very low

[5 rows x 85 columns]
```

[43]:
```
test.head()
```

[43]:
```
          UID  SUMLEVEL  COUNTYID  STATEID         state state_ab  \
27321  255504       140       163       26      Michigan       MI
27322  252676       140         1       23         Maine       ME
27323  276314       140        15       42  Pennsylvania       PA
27324  248614       140       231       21      Kentucky       KY
27325  286865       140       355       48         Texas       TX

                 city                   place     type primary  …  married  \
27321         Detroit  Dearborn Heights City      CDP   tract  …  0.28217
27322          Auburn             Auburn City     City   tract  …  0.64221
27323       Pine City               Millerton  Borough   tract  …  0.59961
27324      Monticello         Monticello City     City   tract  …  0.56953
27325  Corpus Christi                   Edroy     Town   tract  …  0.57620

       married_snp  separated  divorced  split  bad_debt  good_debt  \
27321      0.05910    0.03813   0.14299   Test   0.07651    0.55973
27322      0.02338    0.00000   0.13377   Test   0.14375    0.50380
27323      0.01746    0.01358   0.10026   Test   0.06744    0.38651
27324      0.05492    0.04694   0.12489   Test   0.01741    0.40174
27325      0.01726    0.00588   0.16379   Test   0.03440    0.59748

       population_density  median_age  pop_bins
27321            0.001260   31.189053  very low
27322            0.000257   46.382991  very low
27323            0.000015   43.147420  very low
27324            0.000005   45.155104  very low
27325            0.000452   43.235983  very low

[5 rows x 85 columns]
```

0.2 Data Modelling

[44]:
```
train.columns
```

[44]:
```
Index(['UID', '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', 'split', 'bad_debt', 'good_debt', 'population_density',
       'median_age', 'pop_bins'],
      dtype='object')
```

[45]:
```
train.head()
```

[45]:

	UID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	\
0	267822	140	53	36	New York	NY	Hamilton	
1	246444	140	141	18	Indiana	IN	South Bend	
2	245683	140	63	18	Indiana	IN	Danville	
3	279653	140	127	72	Puerto Rico	PR	San Juan	
4	247218	140	161	20	Kansas	KS	Manhattan	

	place	type	primary	…	married	married_snp	separated	\
0	Hamilton	City	tract	…	0.57851	0.01882	0.01240	
1	Roseland	City	tract	…	0.34886	0.01426	0.01426	
2	Danville	City	tract	…	0.64745	0.02830	0.01607	
3	Guaynabo	Urban	tract	…	0.47257	0.02021	0.02021	
4	Manhattan City	City	tract	…	0.12356	0.00000	0.00000	

	divorced	split	bad_debt	good_debt	population_density	median_age	\
0	0.08770	Train	0.09408	0.43555	0.000026	44.667430	
1	0.09030	Train	0.04274	0.56581	0.001687	34.722748	
2	0.10657	Train	0.09512	0.63972	0.000099	41.774472	
3	0.10106	Train	0.01086	0.51628	0.002442	49.879012	

```
4   0.03109  Train   0.05426    0.46512           0.002207   21.965629

   pop_bins
0  very low
1  very low
2  very low
3  very low
4  very low

[5 rows x 85 columns]
```

[46]:
```python
train['type'].unique()
```

[46]:
```
array(['City', 'Urban', 'Town', 'CDP', 'Village', 'Borough'], dtype=object)
```

[47]:
```python
type_dict={'type':{'City':1, 'Urban':2, 'Town':3, 'CDP':4, 'Village':5,␣
 ↪'Borough':6}}
train.replace(type_dict,inplace=True)
```

```
C:\Users\nilesh\AppData\Local\Temp\ipykernel_8940\1775225308.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  train.replace(type_dict,inplace=True)
```

[48]:
```python
test.replace(type_dict,inplace=True)
```

```
C:\Users\nilesh\AppData\Local\Temp\ipykernel_8940\2850720575.py:1:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  test.replace(type_dict,inplace=True)
```

[49]:
```python
train['type'].unique()
```

[49]:
```
array([1, 2, 3, 4, 5, 6], dtype=int64)
```

[50]:
```python
test['type'].unique()
```

[50]:
```
array([4, 1, 6, 3, 5, 2], dtype=int64)
```

[51]:
```
feature_cols=['COUNTYID','STATEID','zip_code','type','pop',␣
↪'family_mean','second_mortgage', 'home_equity', 'debt','hs_degree',
              'pct_own', 'married','separated', 'divorced']
```

[52]:
```
X_train = train[feature_cols]
y_train = train['hc_mortgage_mean']
```

[53]:
```
X_test = test[feature_cols]
y_test = test['hc_mortgage_mean']
```

[54]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score,␣
↪mean_absolute_error,mean_squared_error,accuracy_score
```

[55]:
```
sc = StandardScaler()
X_train_scaled = sc.fit_transform(X_train)
X_test_scaled = sc.fit_transform(X_test)
```

[56]:
```
lr = LinearRegression()
lr.fit(X_train_scaled, y_train)
```

[56]:
```
LinearRegression()
```

[57]:
```
y_pred= lr.predict(X_test_scaled)
```

[58]:
```
print(y_pred)
```

```
[ 926.77522    1618.71391735 1076.71572816 … 1920.8385097  1467.59345007
 1132.69658305]
```

[59]:
```
r2_score(y_test,y_pred)
```

[59]:
```
0.7381882934134452
```

[60]:
```
mean_absolute_error(y_test, y_pred)
```

[60]:
```
233.86965694140085
```

[61]:
```
mean_squared_error(y_test, y_pred)
```

[61]:
```
103818.40486733473
```

[62]:
```
r2_score(y_train, lr.predict(X_train_scaled))
```

[62]:
```
0.734344756627955
```

Run another model at State level

[63]:
```
state = train['STATEID'].unique()
state
```

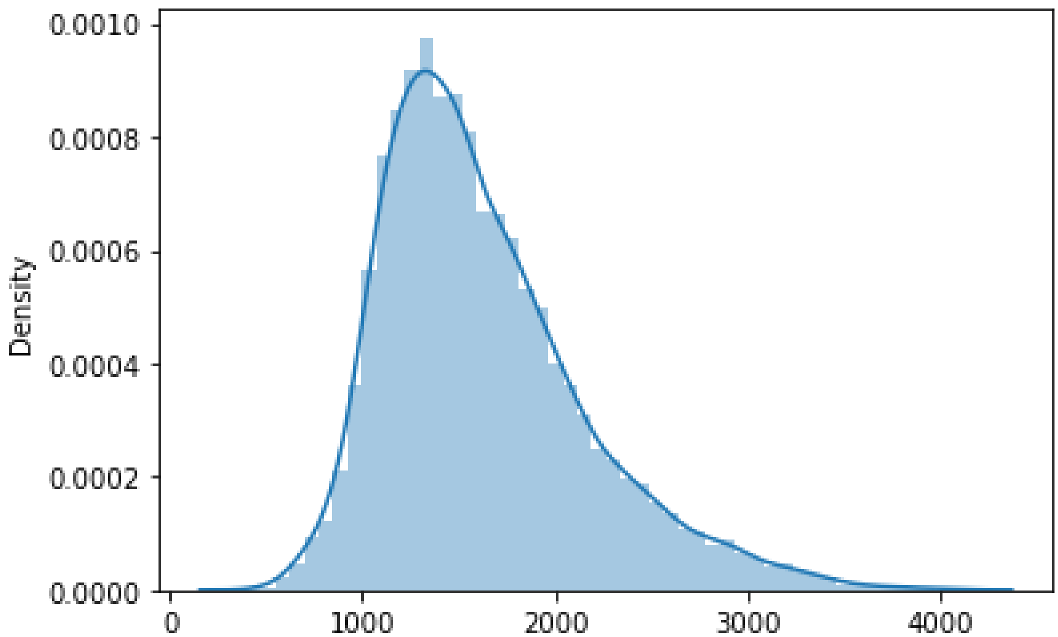
[63]:
```
array([36, 18, 72, 20,  1, 48, 45,  6,  5, 24, 17, 19, 47, 32, 22,  8, 44,
       28, 34, 41,  4, 12, 55, 42, 37, 51, 26, 39, 40, 13, 16, 46, 27, 29,
       53, 56,  9, 54, 21, 25, 11, 15, 30,  2, 33, 49, 50, 31, 38, 35, 23,
       10], dtype=int64)
```

[64]:
```
for i in [20,37,21]:
    print("State ID: ",i)

    X_train_nation = train[train['COUNTYID'] == i][feature_cols]
    y_train_nation = train[train['COUNTYID'] == i]['hc_mortgage_mean']

    X_test_nation = test[test['COUNTYID'] == i][feature_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)

    lr.fit(X_train_scaled_nation,y_train_nation)
    y_pred_nation = lr.predict(X_test_scaled_nation)

    print("Overall R2 score of linear regression model for state ",i,":-" ␣
↪,r2_score(y_test_nation,y_pred_nation))
    print("Overall RMSE of linear regression model for state ",i,":-" ,np.
↪sqrt(mean_squared_error(y_test_nation,y_pred_nation)))
    print("\n")
```

```
State ID:  20
Overall R2 score of linear regression model for state  20 :- 0.6724418142202581
Overall RMSE of linear regression model for state  20 :- 298.8086568658606


State ID:  37
Overall R2 score of linear regression model for state  37 :- 0.707194464316816
Overall RMSE of linear regression model for state  37 :- 337.6896198173383


State ID:  21
Overall R2 score of linear regression model for state  21 :- 0.850093895088195
Overall RMSE of linear regression model for state  21 :- 258.5746925589411
```

[65]:
```
sns.distplot(y_pred)
plt.show()
```

```
C:\Users\nilesh\anaconda3\lib\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)
```



[68]:
```
train.to_csv('train.csv')
test.to_csv('test.csv')
```

[ ]:
```

```