Revised book

August 14, 2023

Import libraries ### Import libraries

[1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

0.0.1 1. Import datasets

[2]:
```
train_df=pd.read_csv('train.csv')
test_df=pd.read_csv('test.csv')
```

[3]:
```
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]
```

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

UID is a primary key and there is no need of indexing

0.0.3 3. 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.

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

[5]: (12513, 80)

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

[6]: (8937, 80)

[7]:
```python
train_df.columns
```

[7]:
```
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')
```

[8]:
```python
for i in range(0, len(np.array_split(train_df.isnull().sum(), 5))):
    print((np.array_split(train_df.isnull().sum(), 5)[i]))
    print()
```

```
UID             0
BLOCKID     12513
SUMLEVEL        0
COUNTYID        0
STATEID         0
state           0
state_ab        0
city            0
place           0
```

```
type              0
primary           0
zip_code          0
area_code         0
lat               0
lng               0
ALand             1
dtype: int64

AWater                1
pop                   1
male_pop              1
female_pop            1
rent_mean           141
rent_median         141
rent_stdev          141
rent_sample_weight  141
rent_samples        141
rent_gt_10          141
rent_gt_15          141
rent_gt_20          141
rent_gt_25          141
rent_gt_30          141
rent_gt_35          141
rent_gt_40          141
dtype: int64

rent_gt_50            141
universe_samples        1
used_samples            1
hi_mean               125
hi_median             125
hi_stdev              125
hi_sample_weight      125
hi_samples            125
family_mean           141
family_median         141
family_stdev          141
family_sample_weight  141
family_samples        141
hc_mortgage_mean      266
hc_mortgage_median    266
hc_mortgage_stdev     266
dtype: int64

hc_mortgage_sample_weight    266
hc_mortgage_samples          266
hc_mean                      291
```

```
hc_median                      291
hc_stdev                       291
hc_samples                     291
hc_sample_weight               291
home_equity_second_mortgage    216
second_mortgage                216
home_equity                    216
debt                           216
second_mortgage_cdf            216
home_equity_cdf                216
debt_cdf                       216
hs_degree                       91
hs_degree_male                  93
dtype: int64

hs_degree_female            109
male_age_mean                90
male_age_median              90
male_age_stdev               90
male_age_sample_weight       90
male_age_samples             90
female_age_mean             101
female_age_median           101
female_age_stdev            101
female_age_sample_weight    101
female_age_samples          101
pct_own                     125
married                      91
married_snp                  91
separated                    91
divorced                     91
dtype: int64
```

[9]:
```python
Fill_rate=(train_df.isnull().sum()/len(train_df))*100
```

[10]:
```python
for i in range(0, len(np.array_split(Fill_rate, 5))):
    print((np.array_split(Fill_rate, 5)[i]))
    print()
```

```
UID           0.000000
BLOCKID     100.000000
SUMLEVEL      0.000000
COUNTYID      0.000000
STATEID       0.000000
state         0.000000
state_ab      0.000000
city          0.000000
```

```
place          0.000000
type           0.000000
primary        0.000000
zip_code       0.000000
area_code      0.000000
lat            0.000000
lng            0.000000
ALand          0.007992
dtype: float64

AWater                 0.007992
pop                    0.007992
male_pop               0.007992
female_pop             0.007992
rent_mean              1.126828
rent_median            1.126828
rent_stdev             1.126828
rent_sample_weight     1.126828
rent_samples           1.126828
rent_gt_10             1.126828
rent_gt_15             1.126828
rent_gt_20             1.126828
rent_gt_25             1.126828
rent_gt_30             1.126828
rent_gt_35             1.126828
rent_gt_40             1.126828
dtype: float64

rent_gt_50               1.126828
universe_samples         0.007992
used_samples             0.007992
hi_mean                  0.998961
hi_median                0.998961
hi_stdev                 0.998961
hi_sample_weight         0.998961
hi_samples               0.998961
family_mean              1.126828
family_median            1.126828
family_stdev             1.126828
family_sample_weight     1.126828
family_samples           1.126828
hc_mortgage_mean         2.125789
hc_mortgage_median       2.125789
hc_mortgage_stdev        2.125789
dtype: float64

hc_mortgage_sample_weight       2.125789
hc_mortgage_samples             2.125789
```

```
hc_mean                        2.325581
hc_median                      2.325581
hc_stdev                       2.325581
hc_samples                     2.325581
hc_sample_weight               2.325581
home_equity_second_mortgage    1.726205
second_mortgage                1.726205
home_equity                    1.726205
debt                           1.726205
second_mortgage_cdf            1.726205
home_equity_cdf                1.726205
debt_cdf                       1.726205
hs_degree                      0.727244
hs_degree_male                 0.743227
dtype: float64

hs_degree_female            0.871094
male_age_mean               0.719252
male_age_median             0.719252
male_age_stdev              0.719252
male_age_sample_weight      0.719252
male_age_samples            0.719252
female_age_mean             0.807161
female_age_median           0.807161
female_age_stdev            0.807161
female_age_sample_weight    0.807161
female_age_samples          0.807161
pct_own                     0.998961
married                     0.727244
married_snp                 0.727244
separated                   0.727244
divorced                    0.727244
dtype: float64
```

BLOCKID has 100% null values, so drop this column.

[11]:
```
train_df.drop(columns=['BLOCKID'],axis=1,inplace=True)
```

[12]:
```
train_df.shape
```

[12]:
```
(12513, 79)
```

[13]:
```
len(train_df.columns[train_df.isnull().sum(axis=0)>0])
```

[13]:
```
65
```

[14]:
```
null_rows=train_df[train_df.isnull().any(axis=1)]
```

[15]:

```
null_rows
```

[15]:

```
          UID  SUMLEVEL  COUNTYID  STATEID        state state_ab  \
51     223593       140        19        4      Arizona       AZ
94     233040       140       101        8     Colorado       CO
153    263292       140        13       34   New Jersey       NJ
302    267158       140        47       36     New York       NY
340    292484       140        25       55    Wisconsin       WI
…         …         …         …        …          …        …
12338  279610       140       127       72  Puerto Rico       PR
12361  274458       140       109       40     Oklahoma       OK
12435  290374       140       710       51     Virginia       VA
12494  246025       140        95       18      Indiana       IN
12512  256921       140       137       27    Minnesota       MN

                city             place   type primary  …  female_age_mean  \
51            Tucson        Littletown    CDP   tract  …         40.02370
94            Pueblo       Pueblo City   City   tract  …         20.00784
153           Newark       Silver Lake   City   tract  …         35.47667
302         Brooklyn     New York City   City   tract  …              NaN
340          Madison      Madison City   City   tract  …         22.03226
…                …                 …      …     …    …              …
12338       San Juan          San Juan  Urban   tract  …         26.77626
12361  Oklahoma City Oklahoma City City    CDP   tract  …         59.38249
12435        Norfolk      Norfolk City   Town   tract  …              NaN
12494      Pendleton         Pendleton   City   tract  …         54.28123
12512         Duluth       Duluth City   City   tract  …              NaN

       female_age_median  female_age_stdev  female_age_sample_weight  \
51              40.83333           8.49563                  30.01695
94              19.25000           4.30291                 172.56153
153             35.58333          20.62717                 369.61740
302                  NaN               NaN                       NaN
340             21.08333           5.13435                1365.86300
…                    …                 …                        …
12338           24.41667          19.03316                 366.92156
12361           64.16667          13.96468                  20.66249
12435                NaN               NaN                       NaN
12494           54.25000           2.78274                   1.67797
12512                NaN               NaN                       NaN

       female_age_samples  pct_own  married  married_snp  separated  divorced
51                  161.0      NaN  0.16308      0.16308    0.02634   0.20499
94                  309.0  0.00000  0.00000      0.00000    0.00000   0.00000
153                1671.0  0.24002  0.37411      0.05579    0.02504   0.07654
302                   NaN      NaN      NaN          NaN        NaN       NaN
340                1981.0  0.00000  0.00773      0.00000    0.00000   0.01160
```

```
...                  ...      ...      ...        ...      ...      ...
12338             1432.0  0.00000  0.03865    0.00000  0.00000  0.05314
12361               67.0  0.02198  0.11712    0.04505  0.00000  0.48649
12435                NaN      NaN      NaN        NaN      NaN      NaN
12494                9.0  0.00000  0.10288    0.10288  0.02337  0.25677
12512                NaN      NaN      NaN        NaN      NaN      NaN

[348 rows x 79 columns]
```

[16]:
```
(348/21450)*100
```

[16]:
```
1.6223776223776225
```

Since only 1.62% of data is missing, we can remove these rows without loosing any information.

[17]:
```
train_df = pd.concat([train_df, null_rows, null_rows]).
 ↪drop_duplicates(keep=False)
```

[18]:
```
train_df.shape
```

[18]:
```
(12165, 79)
```

[19]:
```
len(train_df.columns[train_df.isnull().sum(axis=0)>0])
```

[19]:
```
0
```

[20]:
```
for i in range(0, len(train_df.columns), 10):
    print(train_df[train_df.columns[i:i+10]].head())
    print()
```

```
      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
0        Hamilton   City   tract
1        Roseland   City   tract
2        Danville   City   tract
3        Guaynabo  Urban   tract
4  Manhattan City   City   tract

   zip_code  area_code        lat        lng        ALand     AWater     pop  \
0     13346        315  42.840812 -75.501524  202183361.0  1699120.0  5230.0
1     46616        574  41.701441 -86.266614    1560828.0   100363.0  2633.0
2     46122        317  39.792202 -86.515246   69561595.0   284193.0  6881.0
```

```
3      927        787  18.396103 -66.104169    1105793.0        0.0  2700.0
4    66502        785  39.195573 -96.569366    2554403.0        0.0  5637.0

   male_pop  female_pop  rent_mean
0    2612.0      2618.0  769.38638
1    1349.0      1284.0  804.87924
2    3643.0      3238.0  742.77365
3    1141.0      1559.0  803.42018
4    2586.0      3051.0  938.56493

   rent_median  rent_stdev  rent_sample_weight  rent_samples  rent_gt_10  \
0        784.0   232.63967           272.34441         362.0     0.86761
1        848.0   253.46747           312.58622         513.0     0.97410
2        703.0   323.39011           291.85520         378.0     0.95238
3        782.0   297.39258           259.30316         368.0     0.94693
4        881.0   392.44096          1005.42886        1704.0     0.99286

   rent_gt_15  rent_gt_20  rent_gt_25  rent_gt_30  rent_gt_35
0     0.79155     0.59155     0.45634     0.42817     0.18592
1     0.93227     0.69920     0.69920     0.55179     0.41235
2     0.88624     0.79630     0.66667     0.39153     0.39153
3     0.87151     0.69832     0.61732     0.51397     0.46927
4     0.98247     0.91688     0.84740     0.78247     0.60974

   rent_gt_40  rent_gt_50  universe_samples  used_samples      hi_mean  \
0     0.15493     0.12958             387.0         355.0  63125.28406
1     0.39044     0.27888             542.0         502.0  41931.92593
2     0.28307     0.15873             459.0         378.0  84942.68317
3     0.35754     0.32961             438.0         358.0  48733.67116
4     0.55455     0.44416            1725.0        1540.0  31834.15466

   hi_median     hi_stdev  hi_sample_weight  hi_samples  family_mean
0    48120.0  49042.01206        1290.96240      2024.0  67994.14790
1    35186.0  31639.50203         838.74664      1127.0  50670.10337
2    74964.0  56811.62186        1155.20980      2488.0  95262.51431
3    37845.0  45100.54010         928.32193      1267.0  56401.68133
4    22497.0  34046.50907        1548.67477      1983.0  54053.42396

   family_median  family_stdev  family_sample_weight  family_samples  \
0        53245.0   47667.30119             884.33516          1491.0
1        43023.0   34715.57548             375.28798           554.0
2        85395.0   49292.67664             709.74925          1889.0
3        44399.0   41082.90515             490.18479           729.0
4        50272.0   39609.12605             244.08903           395.0

   hc_mortgage_mean  hc_mortgage_median  hc_mortgage_stdev  \
0        1414.80295              1223.0          641.22898
1         864.41390               784.0          482.27020
```

```
2      1506.06758             1361.0         731.89394
3      1175.28642             1101.0         428.98751
4      1192.58759             1125.0         327.49674

   hc_mortgage_sample_weight  hc_mortgage_samples    hc_mean
0                  377.83135                867.0  570.01530
1                  316.88320                356.0  351.98293
2                  699.41354               1491.0  556.45986
3                  261.28471                437.0  288.04047
4                   76.61052                134.0  443.68855

   hc_median   hc_stdev  hc_samples  hc_sample_weight  \
0      558.0  270.11299       770.0         499.29293
1      336.0  125.40457       229.0         189.60606
2      532.0  184.42175       538.0         323.35354
3      247.0  185.55887       392.0         314.90566
4      444.0   76.12674       124.0          79.55556

   home_equity_second_mortgage  second_mortgage  home_equity     debt  \
0                      0.01588          0.02077      0.08919  0.52963
1                      0.02222          0.02222      0.04274  0.60855
2                      0.00000          0.00000      0.09512  0.73484
3                      0.01086          0.01086      0.01086  0.52714
4                      0.05426          0.05426      0.05426  0.51938

   second_mortgage_cdf  home_equity_cdf
0              0.43658          0.49087
1              0.42174          0.70823
2              1.00000          0.46332
3              0.53057          0.82530
4              0.18332          0.65545

   debt_cdf  hs_degree  hs_degree_male  hs_degree_female  male_age_mean  \
0   0.73341    0.89288         0.85880           0.92434       42.48574
1   0.58120    0.90487         0.86947           0.94187       34.84728
2   0.28704    0.94288         0.94616           0.93952       39.38154
3   0.73727    0.91500         0.90755           0.92043       48.64749
4   0.74967    1.00000         1.00000           1.00000       26.07533

   male_age_median  male_age_stdev  male_age_sample_weight  male_age_samples  \
0         44.00000        22.97306               696.42136            2612.0
1         32.00000        20.37452               323.90204            1349.0
2         40.83333        22.89769               888.29730            3643.0
3         48.91667        23.05968               274.98956            1141.0
4         22.41667        11.84399              1296.89877            2586.0

   female_age_mean
0         44.48629
```

```
1         36.48391
2         42.15810
3         47.77526
4         24.17693

   female_age_median  female_age_stdev  female_age_sample_weight  \
0           45.33333          22.51276                 685.33845
1           37.58333          23.43353                 267.23367
2           42.83333          23.94119                 707.01963
3           50.58333          24.32015                 362.20193
4           21.58333          11.10484                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
```

[21]:
```
train_df['SUMLEVEL'].unique()
```

[21]:
```
array([140])
```

[22]:
```
train_df['primary'].unique()
```

[22]:
```
array(['tract'], dtype=object)
```

'primary' and 'SUMLEVEL' columns has no variane, hence drop these columns from dataset.

[23]:
```
train_df.drop(['SUMLEVEL', 'primary'], axis=1, inplace=True)
```

[24]:
```
train_df.shape
```

[24]:
```
(12165, 77)
```

0.0.4 4. Understanding homeowner costs are incredibly valuable because it is positively correlated to consumer spending which drives the economy through disposable income. Perform debt analysis. You may want to follow the following steps:

- Explore the top 2,500 locations where the percentage of households with a second mortgage is the highest and percent ownership is above 10%. Visualize using geo-map. You may keep the upper limit for the percent of households with a second mortgage to roughly 50%.

[25]:
```
top_2500_locations=train_df[(train_df['second_mortgage']<0.5) &␣
↪(train_df['pct_own']>0.1)].sort_values(by='second_mortgage',ascending=False).
↪head(2500)
```

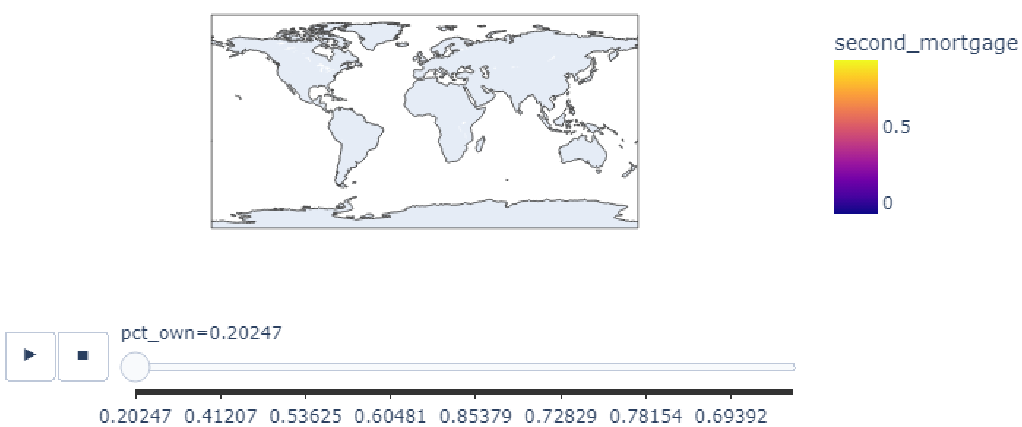
[26]:
```
top_2500_locations
```

[26]:
```
           UID  COUNTYID  STATEID          state state_ab        city  \
11980  251185        27       25  Massachusetts       MA   Worcester
7829   251324         3       24       Maryland       MD  Glen Burnie
2077   235788        57       12        Florida       FL       Tampa
1701   242304        31       17       Illinois       IL     Chicago
11839  242546        31       17       Illinois       IL     Chicago
…         …        …        …            …        …           …
911    239895       135       13        Georgia       GA     Lilburn
7808   267972        55       36       New York       NY   Rochester
6067   273544       153       39           Ohio       OH       Akron
7506   273601       155       39           Ohio       OH      Warren
3118   279940         7       44   Rhode Island       RI  Providence

                    place     type  zip_code  area_code  …  female_age_mean  \
11980      Worcester City     City      1610        508  …         30.60147
7829          Glen Burnie      CDP     21061        410  …         32.53273
2077     Egypt Lake-leto     City     33614        813  …         34.53924
1701          Lincolnwood  Village     60640        773  …         43.85811
11839        Chicago City  Village     60622        773  …         29.46922
…                     …        …         …        …  …              …
911          Lilburn City     City     30047        770  …         38.94562
7808               Greece     City     14616        585  …         41.55050
6067   New Franklin City  Village     44319        330  …         44.02096
7506       Cortland City  Village     44481        330  …         47.12826
3118     Providence City      CDP      2908        401  …         37.78523

       female_age_median  female_age_stdev  female_age_sample_weight  \
11980           26.16667          19.21553                 262.09529
7829            30.66667          19.61959                 694.10357
2077            28.58333          18.56943                 814.45000
1701            39.83333          21.71686                 374.52605
11839           28.50000          17.18452                 449.42977
…                    …                 …                         …
911             41.58333          22.49806                 918.65792
7808            42.50000          23.53709                 666.78464
6067            46.58333          23.66959                 594.26522
7506            48.75000          21.91435                 702.72390
3118            25.83333          22.91600                1428.92915

       female_age_samples  pct_own  married  married_snp  separated  divorced
11980               994.0  0.20247  0.37844      0.11976    0.09341   0.10539
7829               2881.0  0.22380  0.58250      0.08321    0.00000   0.01778
2077               2684.0  0.11618  0.36953      0.12876    0.09957   0.07339
1701               1802.0  0.14228  0.41366      0.13852    0.01771   0.09677
11839              1851.0  0.29468  0.18051      0.00872    0.00872   0.04308
```

```
...                          ...      ...      ...       ...       ...       ...
911                       3500.0  0.86088  0.59390   0.01109   0.00000   0.04437
7808                      2685.0  0.78024  0.48741   0.01432   0.00395   0.08148
6067                      2257.0  0.85329  0.55234   0.01894   0.01210   0.13151
7506                      2939.0  0.82822  0.59251   0.03485   0.02668   0.10413
3118                      3972.0  0.81260  0.39025   0.04663   0.00281   0.03331

[2500 rows x 77 columns]
```

[27]:
```
import plotly.express as px
```

[28]:
```
fig=px.choropleth(top_2500_locations,locations='place',␣
↪color='second_mortgage', scope='world',
                  hover_name='state', animation_frame='pct_own')
fig.show()
```



[29]:
```
import geopandas as gpd
```

[30]:
```
top_2500_locations=top_2500_locations[['second_mortgage','pct_own','zip_code','state','place',
top_2500_locations.head()
```

[30]:
```
       second_mortgage  pct_own  zip_code          state            place  \
11980          0.43363  0.20247      1610  Massachusetts  Worcester City
7829           0.30212  0.22380     21061       Maryland      Glen Burnie
2077           0.28972  0.11618     33614        Florida  Egypt Lake-leto
1701           0.28899  0.14228     60640       Illinois      Lincolnwood
11839          0.27431  0.29468     60622       Illinois     Chicago City

             lat          lng
```

```
11980  42.254262  -71.8003471
7829   39.127273   -76.635265
2077   28.029063   -82.495395
1701   41.967289   -87.652434
11839  41.906640  -87.6895801
```

[31]:
```python
#using geopandas to convert longitude and latitude to points
df_geo=gpd.GeoDataFrame(top_2500_locations,geometry=gpd.
 ↪points_from_xy(top_2500_locations.lng,top_2500_locations.lat))
```

[32]:
```python
df_geo
```

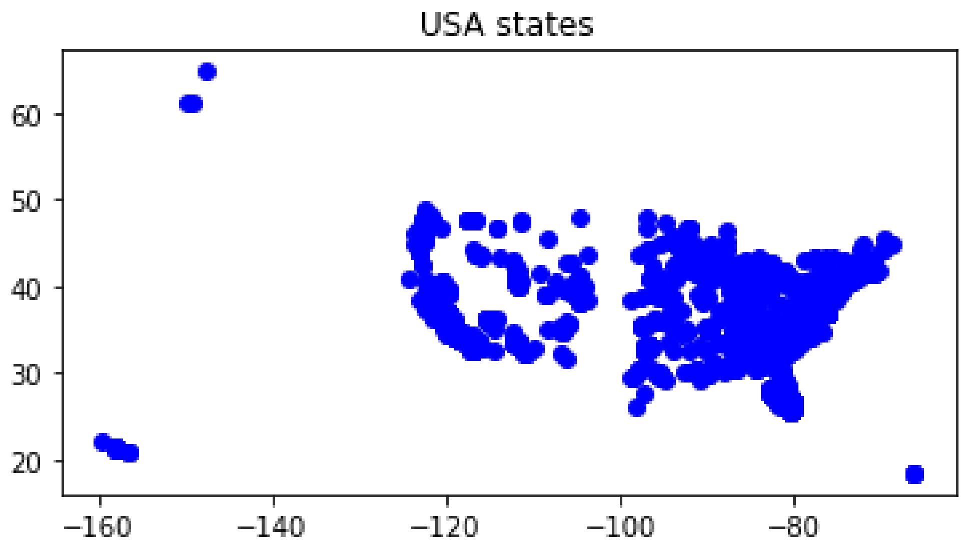
[32]:
```
       second_mortgage  pct_own  zip_code          state              place  \
11980          0.43363  0.20247      1610  Massachusetts     Worcester City
7829           0.30212  0.22380     21061       Maryland        Glen Burnie
2077           0.28972  0.11618     33614        Florida   Egypt Lake-leto
1701           0.28899  0.14228     60640       Illinois        Lincolnwood
11839          0.27431  0.29468     60622       Illinois       Chicago City
…                  …        …        …              …                  …
911            0.04788  0.86088     30047        Georgia       Lilburn City
7808           0.04786  0.78024     14616       New York             Greece
6067           0.04786  0.85329     44319           Ohio  New Franklin City
7506           0.04785  0.82822     44481           Ohio      Cortland City
3118           0.04785  0.81260      2908   Rhode Island   Providence City

             lat          lng                   geometry
11980  42.254262  -71.8003471  POINT (-71.80035 42.25426)
7829   39.127273   -76.635265  POINT (-76.63526 39.12727)
2077   28.029063   -82.495395  POINT (-82.49540 28.02906)
1701   41.967289   -87.652434  POINT (-87.65243 41.96729)
11839  41.906640  -87.6895801  POINT (-87.68958 41.90664)
…              …            …                          …
911    33.871867   -84.112585  POINT (-84.11258 33.87187)
7808   43.242071   -77.652999  POINT (-77.65300 43.24207)
6067   40.969223   -81.554209  POINT (-81.55421 40.96922)
7506   41.319131   -80.772099  POINT (-80.77210 41.31913)
3118   41.843465   -71.450417  POINT (-71.45042 41.84346)

[2500 rows x 8 columns]
```

[33]:
```python
#get built in world dataset
world_data=gpd.read_file(gpd.datasets.get_path('naturalearth_lowres'))
```

[34]:
```python
#plot world map
axis=world_data[world_data.continent=='USA'].plot(
color='lightblue',edgecolor='black')
```

```
df_geo.plot(ax=axis, color='blue')
plt.title('USA states')
```

[34]: Text(0.5, 1.0, 'USA states')



- 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. Bad Debt Equation: Bad Debt = P (Second Mortgage Home Equity Loan) Bad Debt = second_mortgage + home_equity - home_equity_second_mortgage

[35]:
```
train_df['Bad_debt']=train_df['second_mortgage']+train_df['home_equity']
-train_df['home_equity_second_mortgage']
```

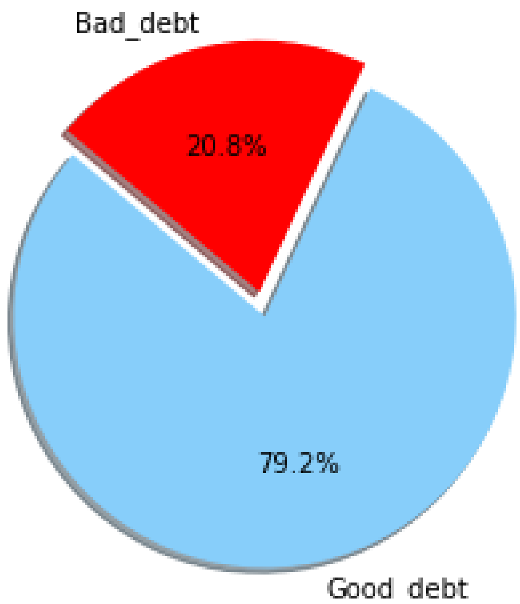
[35]:
```
0        -0.01588
1        -0.02222
2        -0.00000
3        -0.01086
4        -0.05426
           …
12507    -0.02677
12508    -0.00000
12509    -0.00000
12510    -0.06237
12511    -0.01892
Name: home_equity_second_mortgage, Length: 12165, dtype: float64
```

[36]:
```
train_df.columns
```

[36]:
```
Index(['UID', 'COUNTYID', 'STATEID', 'state', 'state_ab', 'city', 'place',
       'type', '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')
```

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

[37]:
```
labels = 'Good_debt', 'Bad_debt'
colors = [ 'lightskyblue','red']
explode = (0.1, 0)
plt.pie([(train_df['debt']-train_df['Bad_debt']).mean()*100,␣
 ↪train_df['Bad_debt'].mean()*100],explode=explode,labels=labels,␣
 ↪colors=colors,
autopct='%1.1f%%', shadow=True, startangle=140)
plt.axis('equal')
plt.show()
```

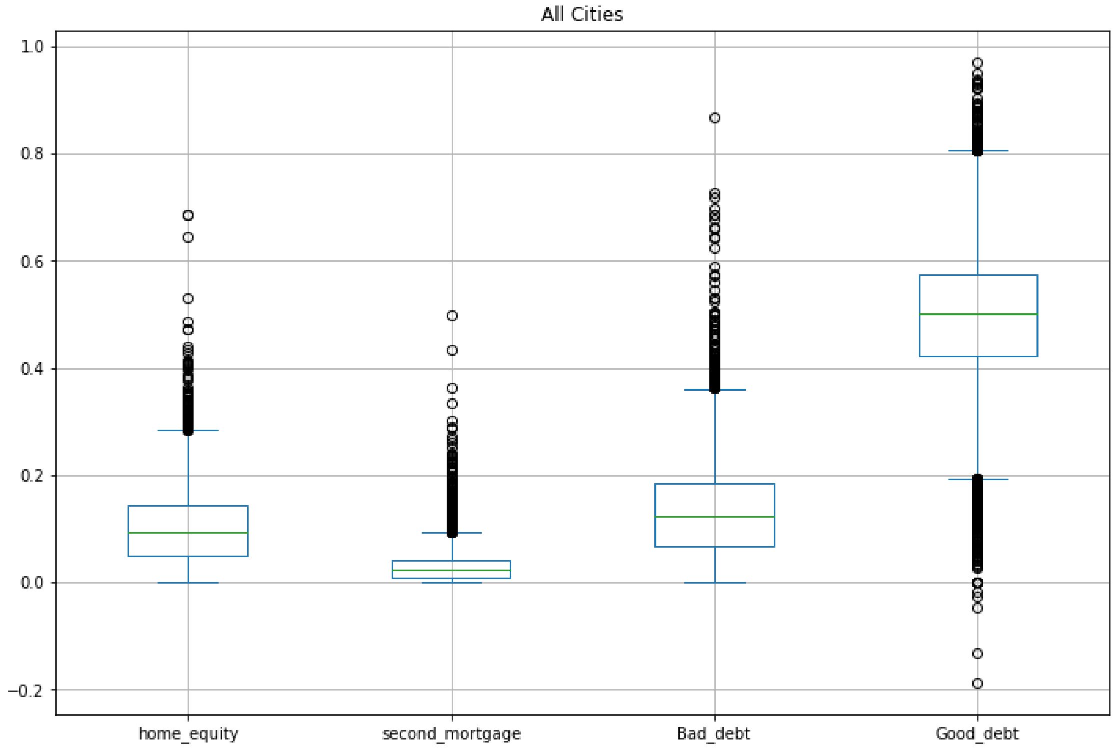


- Create Box and whisker plot and analyze the distribution for 2nd mortgage, home equity, good debt and bad debt for different cities.

[38]:
```
train_df['Good_debt']=train_df['debt']-train_df['Bad_debt']
train_df.columns
```

[38]:
```
Index(['UID', 'COUNTYID', 'STATEID', 'state', 'state_ab', 'city', 'place',
       'type', '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', 'Good_debt'],
      dtype='object')
```

[39]:
```
all_cities = train_df[['home_equity','second_mortgage','Bad_debt', 'Good_debt']]
all_cities.plot.box(figsize=(12,8),grid=True)
plt.title('All Cities')
plt.show()
```

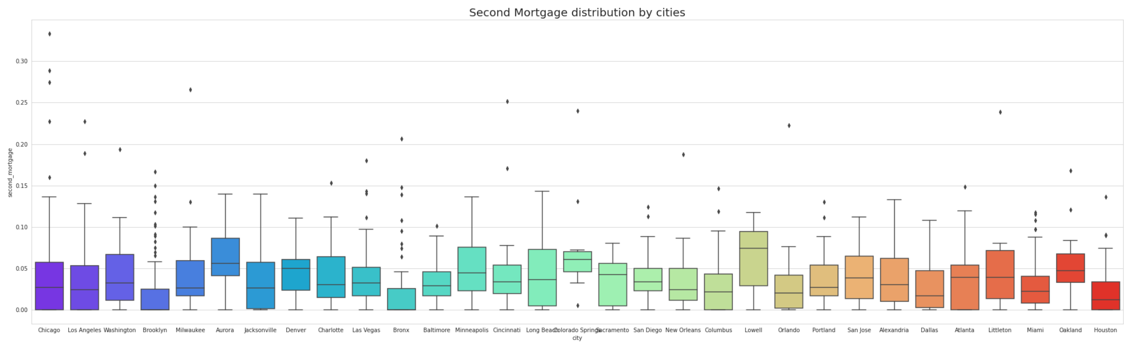


[40]:
```
cities = ['Chicago', 'Los Angeles', 'Washington', 'Brooklyn',
                  'Milwaukee', 'Aurora', 'Jacksonville', 'Denver', 'Charlotte',
                  'Las Vegas', 'Bronx', 'Baltimore', 'Minneapolis',
                  'Cincinnati', 'Long Beach', 'Colorado Springs', 'Sacramento',
                  'San Diego', 'New Orleans', 'Columbus', 'Lowell', 'Orlando',
                  'Portland', 'San Jose', 'Alexandria', 'Dallas', 'Atlanta',
                  'Littleton', 'Miami', 'Oakland', 'Houston']
```

[41]:
```
boxplot_df = train_df[train_df['city'].isin (cities)]
#rpt[rpt['STK_ID'].isin(stk_list)]
```

[42]:
```
sns.set_style("whitegrid")
plt.figure(figsize = (35, 10))
sns.boxplot(x='city',y='second_mortgage',data=boxplot_df,palette='rainbow',␣
 ↪order = ['Chicago', 'Los Angeles', 'Washington', 'Brooklyn',
                  'Milwaukee', 'Aurora', 'Jacksonville', 'Denver', 'Charlotte',
                  'Las Vegas', 'Bronx', 'Baltimore', 'Minneapolis',
                  'Cincinnati', 'Long Beach', 'Colorado Springs', 'Sacramento',
```

```
                     'San Diego', 'New Orleans', 'Columbus', 'Lowell', 'Orlando',
                     'Portland', 'San Jose', 'Alexandria', 'Dallas', 'Atlanta',
                     'Littleton', 'Miami', 'Oakland', 'Houston']).
 ↪set_title('Second Mortgage distribution by cities', fontsize = 20)
plt.show()
```

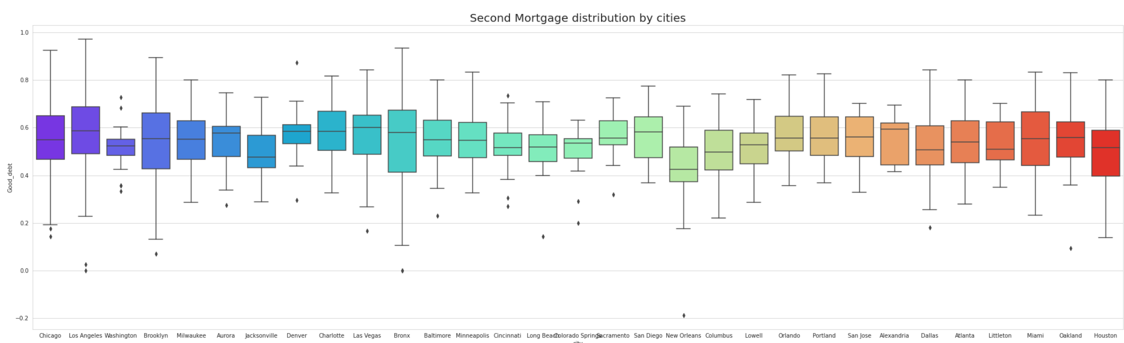


[43]:
```
plt.figure(figsize = (35, 10))
sns.boxplot(x='city',y='home_equity',data=boxplot_df,palette='rainbow', order =␣
 ↪['Chicago', 'Los Angeles', 'Washington', 'Brooklyn',
                     'Milwaukee', 'Aurora', 'Jacksonville', 'Denver', 'Charlotte',
                     'Las Vegas', 'Bronx', 'Baltimore', 'Minneapolis',
                     'Cincinnati', 'Long Beach', 'Colorado Springs', 'Sacramento',
                     'San Diego', 'New Orleans', 'Columbus', 'Lowell', 'Orlando',
                     'Portland', 'San Jose', 'Alexandria', 'Dallas', 'Atlanta',
                     'Littleton', 'Miami', 'Oakland', 'Houston']).
 ↪set_title('Second Mortgage distribution by cities', fontsize = 20)
plt.show()
```

[44]:
```
plt.figure(figsize = (35, 10))
sns.boxplot(x='city',y='Bad_debt',data=boxplot_df,palette='rainbow', order =␣
 ↪['Chicago', 'Los Angeles', 'Washington', 'Brooklyn',
                     'Milwaukee', 'Aurora', 'Jacksonville', 'Denver', 'Charlotte',
```

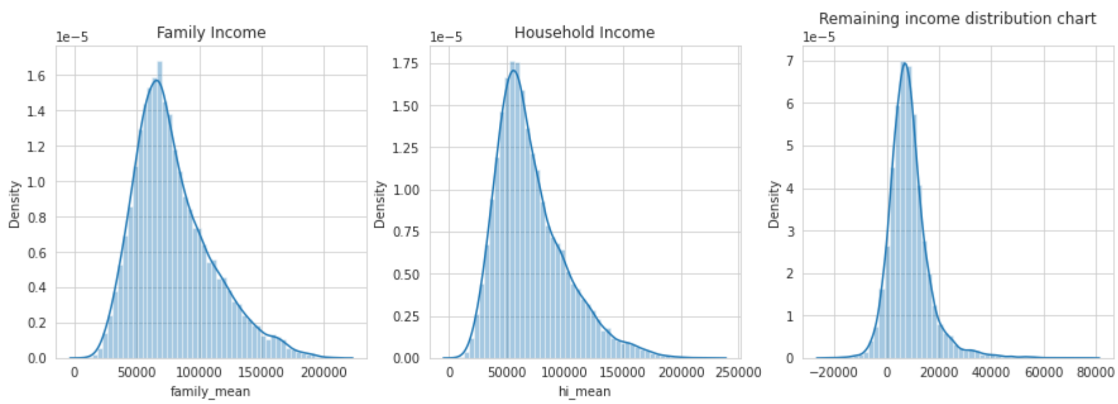
```
                   'Las Vegas', 'Bronx', 'Baltimore', 'Minneapolis',
                   'Cincinnati', 'Long Beach', 'Colorado Springs', 'Sacramento',
                   'San Diego', 'New Orleans', 'Columbus', 'Lowell', 'Orlando',
                   'Portland', 'San Jose', 'Alexandria', 'Dallas', 'Atlanta',
                   'Littleton', 'Miami', 'Oakland', 'Houston']).
 ↪set_title('Second Mortgage distribution by cities', fontsize = 20)
plt.show()
```

[45]:
```
plt.figure(figsize = (35, 10))
sns.boxplot(x='city',y='Good_debt',data=boxplot_df,palette='rainbow', order =␣
 ↪['Chicago', 'Los Angeles', 'Washington', 'Brooklyn',
                   'Milwaukee', 'Aurora', 'Jacksonville', 'Denver', 'Charlotte',
                   'Las Vegas', 'Bronx', 'Baltimore', 'Minneapolis',
                   'Cincinnati', 'Long Beach', 'Colorado Springs', 'Sacramento',
                   'San Diego', 'New Orleans', 'Columbus', 'Lowell', 'Orlando',
                   'Portland', 'San Jose', 'Alexandria', 'Dallas', 'Atlanta',
                   'Littleton', 'Miami', 'Oakland', 'Houston']).
 ↪set_title('Second Mortgage distribution by cities', fontsize = 20)
plt.show()
```



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

[46]:
```python
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()
```



0.0.5 5. Perform EDA and come out with insights into population density and age. You may require deriving new fields (Make sure to weight averages for accurate measurements):

- Population density (hint-use ‘pop’ and ‘ALand’ to calculate)

[47]:
```python
train_df['population_density']=train_df['pop']/train_df['ALand']
```

[48]:
```python
train_df.head()
```

[48]:
```
      UID  COUNTYID  STATEID        state state_ab        city  \
0  267822        53       36     New York       NY    Hamilton
1  246444       141       18      Indiana       IN  South Bend
2  245683        63       18      Indiana       IN    Danville
3  279653       127       72  Puerto Rico       PR    San Juan
4  247218       161       20       Kansas       KS   Manhattan

         place  type  zip_code  area_code  …  female_age_sample_weight  \
0     Hamilton  City     13346        315  …                 685.33845
1     Roseland  City     46616        574  …                 267.23367
```

```
2        Danville   City     46122        317  …                707.01963
3        Guaynabo  Urban       927        787  …                362.20193
4  Manhattan City   City     66502        785  …               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

   Bad_debt  Good_debt  population_density
0   0.10996    0.41967            0.000026
1   0.06496    0.54359            0.001687
2   0.09512    0.63972            0.000099
3   0.02172    0.50542            0.002442
4   0.10852    0.41086            0.002207

[5 rows x 80 columns]
```

- median age (hint-use the variables ‘male_age_median’, ‘female_age_median’, ‘male_pop’, ‘female_pop’)

[49]:

```
train_df['median_age']=((train_df['female_age_median']*train_df['female_pop'])
+(train_df['male_age_median']*train_df['male_pop']))/
 ↪(train_df['female_pop']+train_df['male_pop'])
```

[50]:

```
train_df.head()
```

[50]:

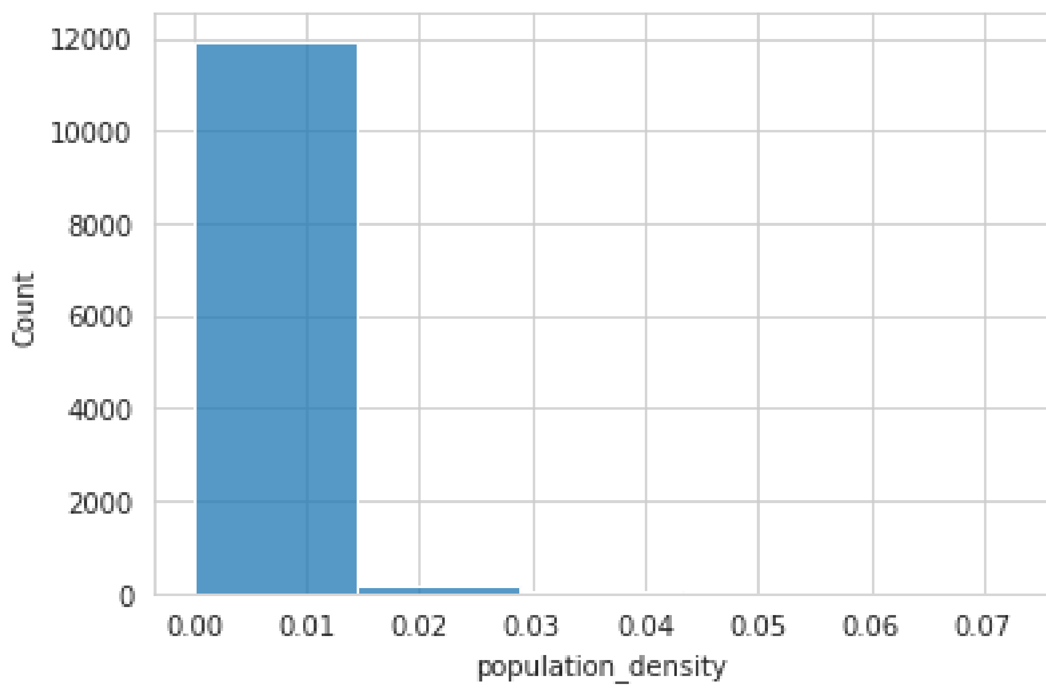
```
      UID  COUNTYID  STATEID        state state_ab        city  \
0  267822        53       36     New York       NY    Hamilton
1  246444       141       18      Indiana       IN  South Bend
2  245683        63       18      Indiana       IN    Danville
3  279653       127       72  Puerto Rico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

           place   type  zip_code  area_code  …  female_age_samples  \
0       Hamilton   City     13346        315  …              2618.0
1       Roseland   City     46616        574  …              1284.0
2       Danville   City     46122        317  …              3238.0
3       Guaynabo  Urban       927        787  …              1559.0
4  Manhattan City   City     66502        785  …              3051.0

   pct_own  married  married_snp  separated  divorced  Bad_debt  Good_debt  \
0  0.79046  0.57851      0.01882    0.01240   0.08770   0.10996    0.41967
1  0.52483  0.34886      0.01426    0.01426   0.09030   0.06496    0.54359
2  0.85331  0.64745      0.02830    0.01607   0.10657   0.09512    0.63972
```

```
3  0.65037  0.47257      0.02021    0.02021   0.10106   0.02172    0.50542
4  0.13046  0.12356      0.00000    0.00000   0.03109   0.10852    0.41086

   population_density  median_age
0            0.000026   44.667430
1            0.001687   34.722748
2            0.000099   41.774472
3            0.002442   49.879012
4            0.002207   21.965629

[5 rows x 81 columns]
```

- Visualize the findings using appropriate chart type.

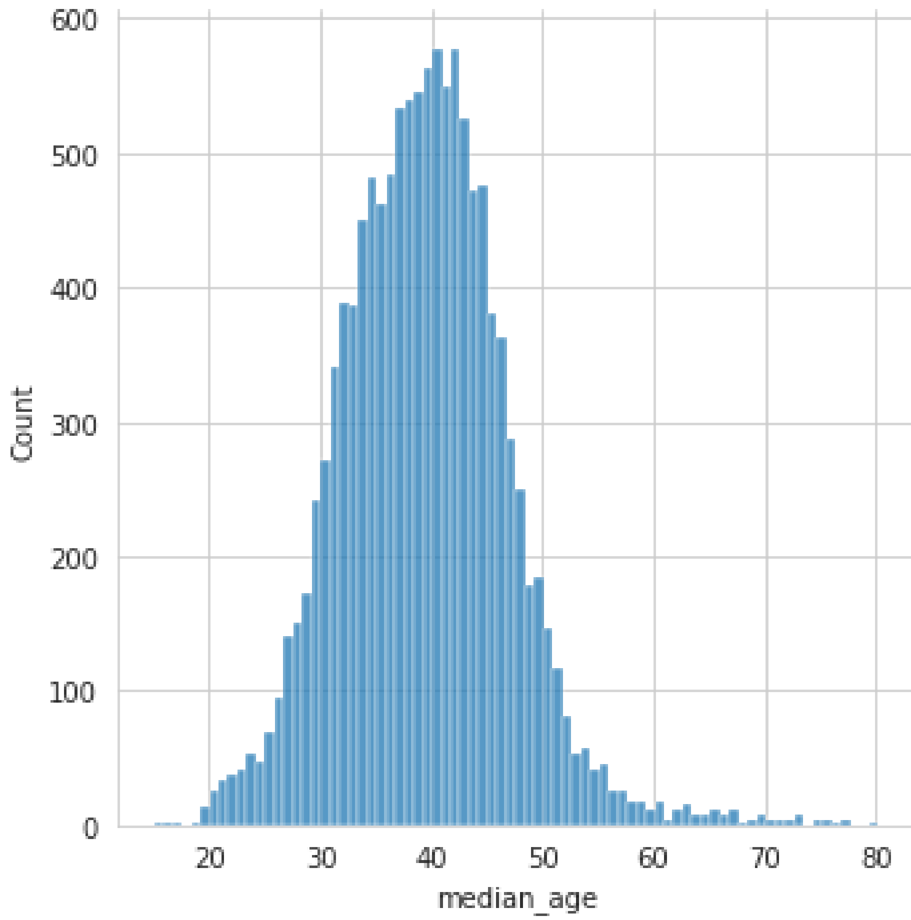
[51]:
```
sns.histplot(train_df['population_density'],bins=5)
```

[51]: <AxesSubplot: xlabel='population_density', ylabel='Count'>



[52]:
```
sns.displot(train_df['median_age'])
```

[52]: <seaborn.axisgrid.FacetGrid at 0x7fb4cfaa2b90>



0.0.6 6. 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.

[53]:
```python
bins = [0, 12,18, 35, 55, 100]
labels = ['kids', 'Youth', 'Young Adult', 'Adult', 'Senior']
```

[54]:
```python
train_df['male_population_bracket'] = pd.cut(train_df['male_age_median'], bins,␣
↪labels = labels)
```

[55]:
```python
train_df['female_population_bracket'] = pd.cut(train_df['female_age_median'],␣
↪bins, labels = labels)
```

[56]:
```python
train_df.columns
```

[56]:
```
Index(['UID', 'COUNTYID', 'STATEID', 'state', 'state_ab', 'city', 'place',
       'type', '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', 'Good_debt', 'population_density', 'median_age',
        'male_population_bracket', 'female_population_bracket'],
       dtype='object')
```

[57]:
```
train_df['male_population_bracket'].value_counts()
```

[57]:
```
Adult          7702
Young Adult    4196
Senior          250
Youth            16
kids              1
Name: male_population_bracket, dtype: int64
```

[58]:
```
train_df['female_population_bracket'].value_counts()
```

[58]:
```
Adult          8831
Young Adult    2951
Senior          380
Youth             3
kids              0
Name: female_population_bracket, dtype: int64
```

[59]:
```
train_df['female_population_bracket']
```

[59]:
```
0              Adult
1              Adult
2              Adult
3              Adult
4        Young Adult
            …
12507          Adult
12508    Young Adult
12509          Adult
```

```
12510    Young Adult
12511          Adult
Name: female_population_bracket, Length: 12165, dtype: category
Categories (5, object): ['kids' < 'Youth' < 'Young Adult' < 'Adult' < 'Senior']
```

[60]:
```python
train_df['married']
```

[60]:
```
0        0.57851
1        0.34886
2        0.64745
3        0.47257
4        0.12356
          …
12507    0.49414
12508    0.29096
12509    0.41594
12510    0.44332
12511    0.53217
Name: married, Length: 12165, dtype: float64
```

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

[61]:
```python
pop_bin_male=train_df.
↪groupby(by='male_population_bracket')[['married','separated','divorced']].
↪mean()
```

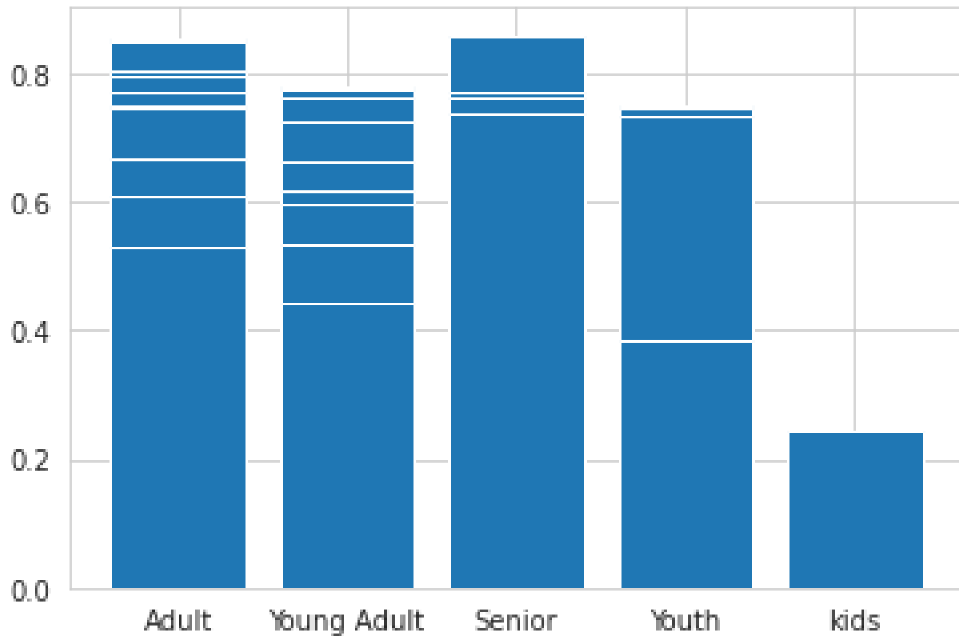
[62]:
```python
pop_bin_male
```

[62]:
```
                          married  separated  divorced
male_population_bracket
kids                     0.244980   0.000000  0.020080
Youth                    0.437084   0.029179  0.062047
Young Adult              0.433805   0.022619  0.093513
Adult                    0.552090   0.017620  0.104026
Senior                   0.629958   0.015045  0.119513
```

- Visualize using appropriate chart type.

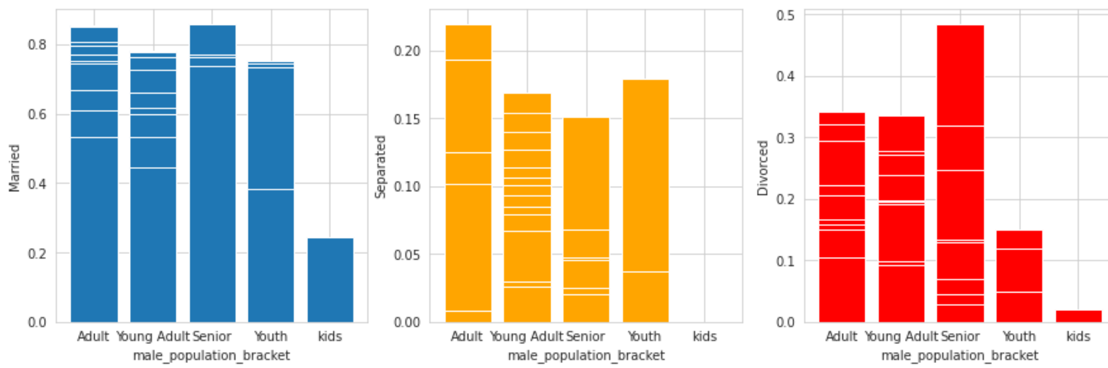
[63]:
```python
plt.bar(train_df['male_population_bracket'],train_df['married'])
```

[63]:
```
<BarContainer object of 12165 artists>
```



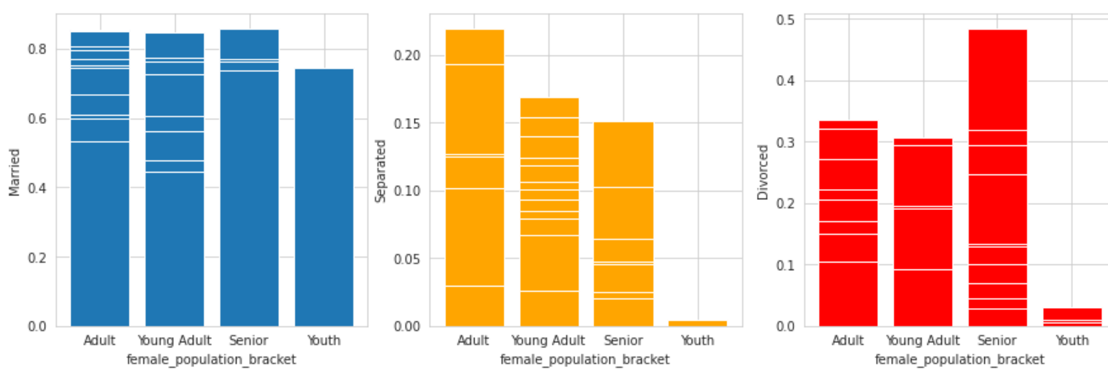
[64]:
```
plt.figure(figsize=(15,10))
plt.subplot(2,3,1)
plt.bar(train_df['male_population_bracket'],train_df['married'])
plt.xlabel('male_population_bracket')
plt.ylabel('Married')
plt.subplot(2,3,2)
plt.
 ↪bar(train_df['male_population_bracket'],train_df['separated'],color='orange')
plt.xlabel('male_population_bracket')
plt.ylabel('Separated')
plt.subplot(2,3,3)
plt.bar(train_df['male_population_bracket'],train_df['divorced'],color='red')
plt.xlabel('male_population_bracket')
plt.ylabel('Divorced')
```

[64]: Text(0, 0.5, 'Divorced')



[65]:
```
plt.figure(figsize=(15,10))
plt.subplot(2,3,1)
plt.bar(train_df['female_population_bracket'],train_df['married'])
plt.xlabel('female_population_bracket')
plt.ylabel('Married')
plt.subplot(2,3,2)
plt.
 ↪bar(train_df['female_population_bracket'],train_df['separated'],color='orange')
plt.xlabel('female_population_bracket')
plt.ylabel('Separated')
plt.subplot(2,3,3)
plt.bar(train_df['female_population_bracket'],train_df['divorced'],color='red')
plt.xlabel('female_population_bracket')
plt.ylabel('Divorced')
```

[65]:
```
Text(0, 0.5, 'Divorced')
```



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

[66]:
```
rent_mean_state=train_df.groupby(by='state')['rent_mean'].mean()
```

[67]:
```
rent_mean_state.head()
```

[67]:
```
state
Alabama        771.042232
Alaska        1156.152927
Arizona       1135.750582
Arkansas       721.914618
California    1489.618355
Name: rent_mean, dtype: float64
```

[68]:
```
overall_income_mean_state=train_df.groupby(by='state')['hi_mean'].mean()
```

[69]:
```
overall_income_mean_state.head()
```

[69]:
```
state
Alabama       57950.893604
Alaska        77778.255351
Arizona       69566.311650
Arkansas      54534.802898
California    82940.215376
Name: hi_mean, dtype: float64
```

[70]:
```
percentage_of_rent=(rent_mean_state/overall_income_mean_state)*100
```

[71]:
```
percentage_of_rent.head()
```

[71]:
```
state
Alabama       1.330510
Alaska        1.486473
Arizona       1.632616
Arkansas      1.323769
California    1.796015
dtype: float64
```

[72]:
```
percentage_of_rent_1=percentage_of_rent.head(10)
```

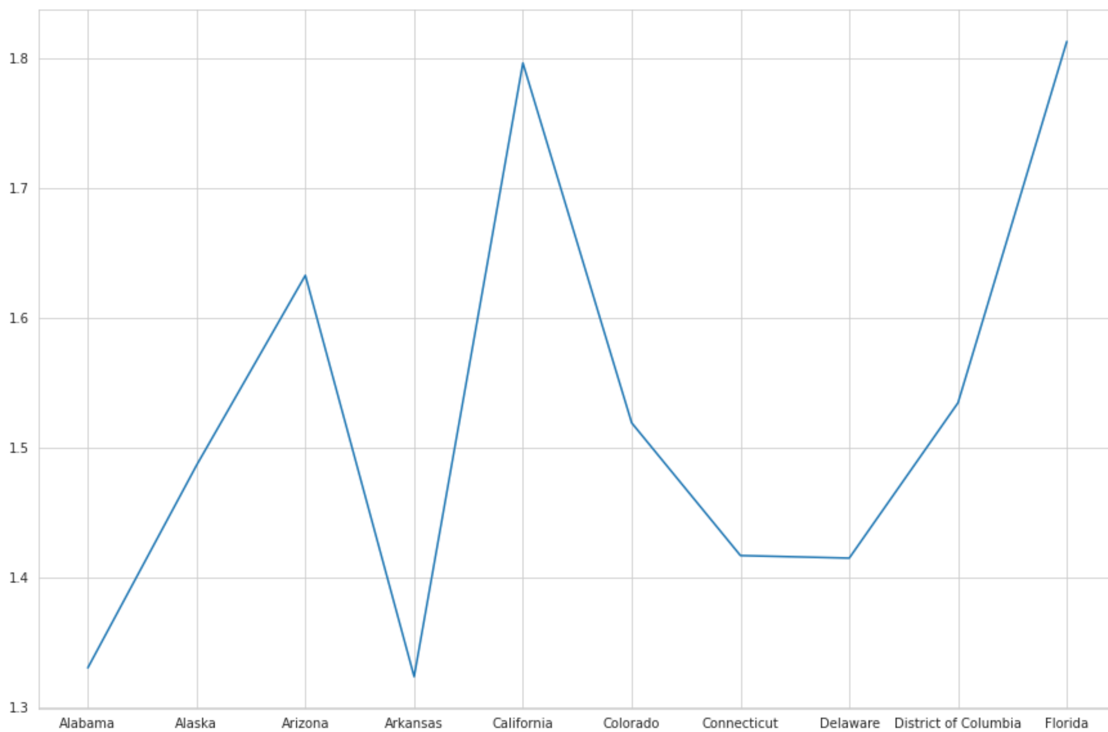
[73]:
```
percentage_of_rent_1
```

[73]:
```
state
Alabama                 1.330510
Alaska                  1.486473
Arizona                 1.632616
Arkansas                1.323769
```

```
California              1.796015
Colorado                1.519084
Connecticut             1.416959
Delaware                1.414990
District of Columbia    1.534523
Florida                 1.812602
dtype: float64
```

[74]:
```
plt.figure(figsize=(15,10))
plt.plot(percentage_of_rent_1)
```

[74]:
```
[<matplotlib.lines.Line2D at 0x7fb495f1fd60>]
```



[75]:
```
overall_percentage_of_rent=(rent_mean_state.sum()/overall_income_mean_state.
↪sum())*100
```

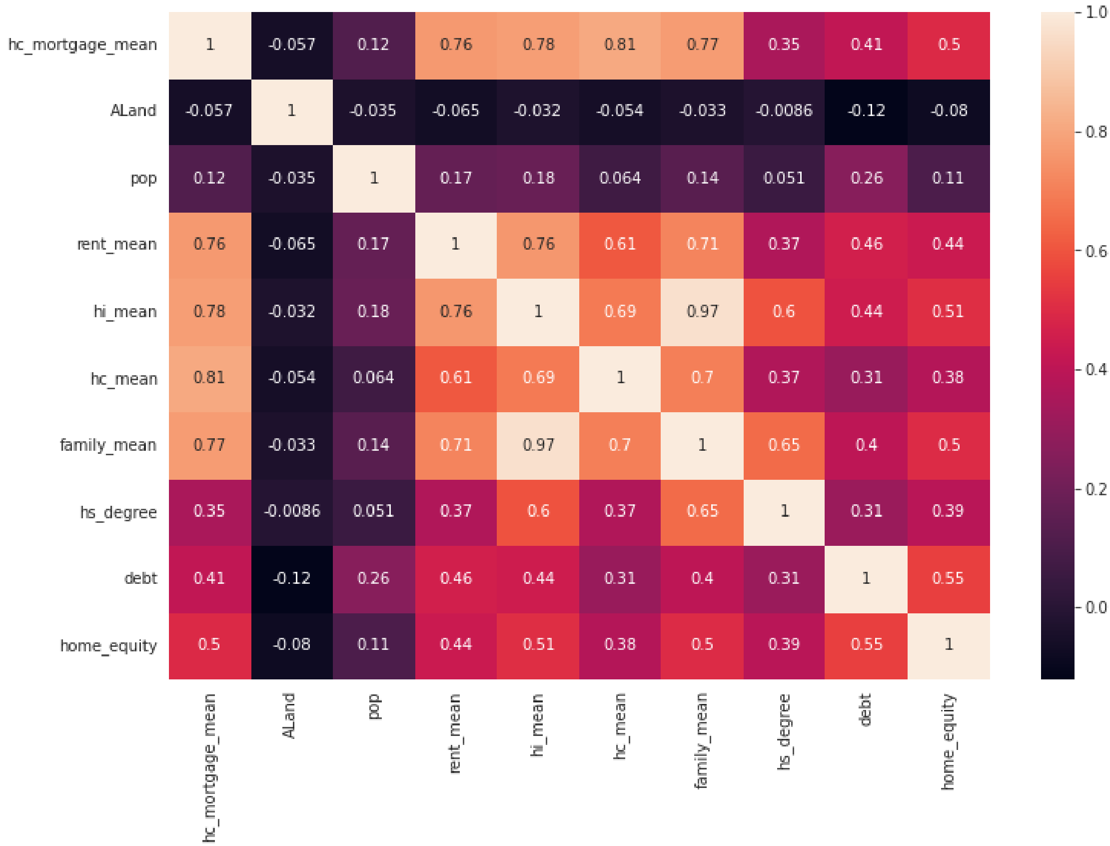
[76]:
```
overall_percentage_of_rent
```

[76]:
```
1.4180239969724684
```

Overall rent as percentage of income at overall level is 1.41%

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

[77]:
```
plt.figure(figsize=(12,8))
sns.
↪heatmap(data=train_df[['hc_mortgage_mean','ALand','pop','rent_mean','hi_mean','hc_mean','fa
                                   'hs_degree','debt','home_equity']].corr(),annot=True)
plt.show()
```



rent_mean, hi_mean, hc_mean, family_mean has a good correlation with the target i.e- hc_mortagage_mean

0.0.9 10. 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 mean monthly mortgage and owner costs of specified geographical location. Note: Exclude loans from prediction model which have NaN values for hc_mortgage_mean. NaN represents not a number/missing values.

1. Run a model at a Nation level. If the accuracy levels and R square are not satisfactory proceed to below step
2. Run another model at State level. There are 52 states in USA.

Considerations: Keep below considerations while building a linear regression model

1. Variables should have significant impact on predicting Monthly mortgage and owner costs
2. Utilize all predictor variable to start with initial hypothesis
3. R square of 60% and above should be achieved
4. Ensure Multi-collinearity does not exist in dependent variables
5. Test if predicted variable is normally distributed

[78]:
```
len(train_df.columns[train_df.isnull().sum(axis=0)>0])
```

[78]:
```
0
```

[79]:
```
train_df.shape
```

[79]:
```
(12165, 83)
```

[80]:
```
test_df.shape
```

[80]:
```
(8937, 80)
```

[81]:
```
test_df.head()
```

[81]:
```
      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]
```

[82]:
```python
len(test_df.columns[test_df.isnull().sum(axis=0)>0])
```

[82]: 59

[83]:
```python
for i in range(0, len(np.array_split(test_df.isnull().sum(), 5))):
    print((np.array_split(test_df.isnull().sum(), 5)[i]))
    print()
```

```
UID            0
BLOCKID     8937
SUMLEVEL       0
COUNTYID       0
STATEID        0
state          0
state_ab       0
city           0
place          0
type           0
primary        0
zip_code       0
area_code      0
lat            0
lng            0
ALand          0
dtype: int64

AWater                   0
pop                      0
male_pop                 0
female_pop               0
rent_mean              111
rent_median            111
rent_stdev             111
rent_sample_weight     111
rent_samples           111
rent_gt_10             111
rent_gt_15             111
rent_gt_20             111
rent_gt_25             111
rent_gt_30             111
rent_gt_35             111
rent_gt_40             111
dtype: int64
```

```
rent_gt_50              111
universe_samples          0
used_samples              0
hi_mean                  90
hi_median                90
hi_stdev                 90
hi_sample_weight         90
hi_samples               90
family_mean             100
family_median           100
family_stdev            100
family_sample_weight    100
family_samples          100
hc_mortgage_mean        201
hc_mortgage_median      201
hc_mortgage_stdev       201
dtype: int64

hc_mortgage_sample_weight      201
hc_mortgage_samples            201
hc_mean                        212
hc_median                      212
hc_stdev                       212
hc_samples                     212
hc_sample_weight               212
home_equity_second_mortgage    162
second_mortgage                162
home_equity                    162
debt                           162
second_mortgage_cdf            162
home_equity_cdf                162
debt_cdf                       162
hs_degree                       61
hs_degree_male                  65
dtype: int64

hs_degree_female            75
male_age_mean               61
male_age_median             61
male_age_stdev              61
male_age_sample_weight      62
male_age_samples            62
female_age_mean             69
female_age_median           69
female_age_stdev            69
female_age_sample_weight    69
female_age_samples          69
```

```
pct_own                    91
married                    62
married_snp                62
separated                  62
divorced                   62
dtype: int64
```

[84]:
```
test_df.drop(columns=['BLOCKID'],axis=1,inplace=True)
```

[85]:
```
test_df.shape
```

[85]:
```
(8937, 79)
```

[86]:
```
null_rows_1=test_df[test_df.isnull().any(axis=1)]
null_rows_1
```

[86]:
```
         UID  SUMLEVEL  COUNTYID  STATEID          state state_ab  \
17    265339       140         3       32         Nevada       NV
27    287596       140       451       48          Texas       TX
44    250903       140        25       25  Massachusetts       MA
54    287557       140       441       48          Texas       TX
70    247510       140       209       20         Kansas       KS
…        …        …        …       …              …       …
8858  284740       140       141       48          Texas       TX
8873  286304       140       245       48          Texas       TX
8882  240129       140       179       13        Georgia       GA
8902  290734       140         7       50        Vermont       VT
8936  225965       140        37        6     California       CA

                   city              place  type primary  …  female_age_mean  \
17            Las Vegas         Winchester  City   tract  …         33.57247
27           San Angelo    San Angelo City  Town   tract  …         21.40298
44            Cambridge     Cambridge City  City   tract  …         22.53871
54              Abilene           Tye City  Town   tract  …         22.72458
70          Kansas City   Kansas City City  City   tract  …              NaN
…                   …                  …    …     …  …              …
8858            El Paso        Fort Bliss  Town   tract  …         39.22728
8873        Port Arthur   Central Gardens  Town   tract  …         16.00833
8882       Fort Stewart       Fort Stewart  City   tract  …         19.95132
8902  South Burlington    Essex Junction   CDP   tract  …              NaN
8936        Los Angeles  Los Angeles City  City   tract  …              NaN

      female_age_median  female_age_stdev  female_age_sample_weight  \
17             32.50000          17.36519                  49.31407
27             20.50000           7.28394                 456.32778
44             20.75000           7.40442                2069.57453
```

```
54           23.16667      2.18207            26.71180
70                NaN          NaN                 NaN
…                 …            …                   …
8858         38.08333     26.40984           332.38824
8873         16.08333      1.19617             1.57143
8882         20.50000     13.65045           814.71125
8902              NaN          NaN                 NaN
8936              NaN          NaN                 NaN

      female_age_samples  pct_own  married  married_snp  separated  divorced
17                 234.0  0.00000  0.22857      0.11020    0.06122   0.26327
27                 868.0  0.00000  0.22232      0.17475    0.01052   0.00000
44                3716.0  0.02169  0.10879      0.05440    0.00204   0.00409
54                  59.0  0.60000  0.01984      0.00933    0.00000   0.02217
70                   NaN      NaN      NaN          NaN        NaN       NaN
…                    …        …       …     …           …          …
8858              1385.0  0.10858  0.44740      0.09055    0.06924   0.07324
8873                 5.0      NaN  0.19881      0.19881    0.04829   0.15871
8882              3424.0  0.00264  0.84226      0.03793    0.00482   0.02830
8902                 NaN      NaN      NaN          NaN        NaN       NaN
8936                 NaN      NaN      NaN          NaN        NaN       NaN

[264 rows x 79 columns]
```

[87]:
```python
(264/8937)*100
```

[87]:
```
2.9540114132259148
```

[88]:
```python
test_df = pd.concat([test_df, null_rows_1, null_rows_1]).
 ↪drop_duplicates(keep=False)
```

[89]:
```python
len(test_df.columns[test_df.isnull().sum(axis=0)>0])
```

[89]:
```
0
```

[90]:
```python
test_df.shape
```

[90]:
```
(8673, 79)
```

[91]:
```python
train_df.columns
```

[91]:
```
Index(['UID', 'COUNTYID', 'STATEID', 'state', 'state_ab', 'city', 'place',
       'type', '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', 'Good_debt', 'population_density', 'median_age',
       'male_population_bracket', 'female_population_bracket'],
      dtype='object')
```

[92]:
```
test_df.columns
```

[92]:
```
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'],
      dtype='object')
```

[93]:
```
test_df.drop(columns=['SUMLEVEL','primary'],axis=1,inplace=True)
```

[94]:
```
test_df.shape
```

[94]:
```
(8673, 77)
```

[95]:
```
train_df.drop(columns=['Bad_debt', 'Good_debt', 'population_density',␣
↪'median_age',
```

```
'male_population_bracket',␣
↪'female_population_bracket'],axis=1,inplace=True)
```

[96]:
```
train_df.shape
```

[96]: (12165, 77)

[97]:
```
print(train_df['hc_mortgage_mean'].isna().sum(), test_df['hc_mortgage_mean'].
↪isna().sum())
```

```
0 0
```

[98]:
```
train_df.columns
```

[98]:
```
Index(['UID', 'COUNTYID', 'STATEID', 'state', 'state_ab', 'city', 'place',
       'type', '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')
```

[99]:
```
test_df.columns
```

[99]:
```
Index(['UID', 'COUNTYID', 'STATEID', 'state', 'state_ab', 'city', 'place',
       'type', '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')
```

[100]:
```
drop_variables=['UID', 'state', 'state_ab', 'city', 'place', 'type',␣
 ↪'zip_code', 'area_code',
                'lat', 'lng']
```

[101]:
```
train_df.drop(drop_variables,axis=1,inplace=True)
```

[102]:
```
test_df.drop(drop_variables,axis=1,inplace=True)
```

[103]:
```
train_df.shape
```

[103]:
```
(12165, 67)
```

[104]:
```
test_df.shape
```

[104]:
```
(8673, 67)
```

[105]:
```
train_df.drop(columns=['COUNTYID', 'STATEID'],axis=1,inplace=True)
```

[106]:
```
test_df.drop(columns=['COUNTYID', 'STATEID'],axis=1,inplace=True)
```

[107]:
```
test_df.shape
```

[107]:
```
(8673, 65)
```

[108]:
```
train_df.shape
```

[108]:
```
(12165, 65)
```

[109]:
```
y_train=train_df['hc_mortgage_mean']
```

[110]:
```
y_train
```

[110]:
```
0        1414.80295
1         864.41390
2        1506.06758
3        1175.28642
4        1192.58759
            …
```

```
12507    1304.01913
12508    1571.95506
12509     935.20082
12510    1576.27493
12511    2578.60244
Name: hc_mortgage_mean, Length: 12165, dtype: float64
```

[111]:
```
x_train=train_df.drop(columns=['hc_mortgage_mean'])
```

[112]:
```
x_train
```

[112]:
```
              ALand      AWater     pop  male_pop  female_pop   rent_mean  \
0      202183361.0  1699120.0  5230.0    2612.0      2618.0   769.38638
1        1560828.0   100363.0  2633.0    1349.0      1284.0   804.87924
2       69561595.0   284193.0  6881.0    3643.0      3238.0   742.77365
3        1105793.0        0.0  2700.0    1141.0      1559.0   803.42018
4        2554403.0        0.0  5637.0    2586.0      3051.0   938.56493
…              …          …       …         …           …
12507    6429449.0     6059.0  3643.0    1619.0      2024.0  1098.77795
12508    1892447.0     2004.0  1903.0     874.0      1029.0   999.97399
12509    6742997.0    27745.0  5674.0    2612.0      3062.0   744.39241
12510    3211646.0    10033.0  6263.0    3084.0      3179.0   926.61929
12511    1493249.0        0.0  5719.0    2851.0      2868.0  1526.05146

       rent_median  rent_stdev  rent_sample_weight  rent_samples  …  \
0            784.0   232.63967           272.34441         362.0  …
1            848.0   253.46747           312.58622         513.0  …
2            703.0   323.39011           291.85520         378.0  …
3            782.0   297.39258           259.30316         368.0  …
4            881.0   392.44096          1005.42886        1704.0  …
…              …          …                   …            …  …
12507       1091.0   212.66221           166.50176         501.0  …
12508        977.0   222.19385           431.22657         986.0  …
12509        722.0   176.91612           662.45941         837.0  …
12510        908.0   312.44578           510.07451         945.0  …
12511       1424.0   634.94982           221.22791         830.0  …

       female_age_mean  female_age_median  female_age_stdev  \
0             44.48629           45.33333          22.51276
1             36.48391           37.58333          23.43353
2             42.15810           42.83333          23.94119
3             47.77526           50.58333          24.32015
4             24.17693           21.58333          11.10484
…                 …                 …                 …
12507         36.17142           38.00000          19.50123
12508         30.87275           25.91667          15.69777
12509         43.45140           46.16667          23.73648
```

```
12510         37.82698           33.25000          24.45782
12511         38.97593           38.83333          20.66153

       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
…                           …                   …      …        …
12507                 456.98078              2024.0  0.64754  0.49414
12508                 315.07146              1029.0  0.08881  0.29096
12509                 804.60756              3062.0  0.50141  0.41594
12510                 764.81692              3179.0  0.58918  0.44332
12511                 704.81384              2868.0  0.67227  0.53217

       married_snp  separated  divorced
0          0.01882    0.01240   0.08770
1          0.01426    0.01426   0.09030
2          0.02830    0.01607   0.10657
3          0.02021    0.02021   0.10106
4          0.00000    0.00000   0.03109
…              …          …       …
12507      0.03953    0.00805   0.15007
12508      0.01835    0.00655   0.08257
12509      0.05894    0.02995   0.09903
12510      0.04636    0.02610   0.09272
12511      0.00046    0.00000   0.10524

[12165 rows x 64 columns]
```

[113]:
```python
print(train_df.shape)
```

```
(12165, 65)
```

[114]:
```python
from sklearn.linear_model import LinearRegression
```

[115]:
```python
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score,␣
↪SCORERS
```

[116]:
```python
def adj_rsqrd(df, r2):
        # adjusted r2 using formula adj_r2 = 1 - (1- r2) * (n-1) / (n - k - 1)
    # k = number of predictors = data.shape[1] - 1
    adj_rsqrd = 1 - (1-r2)*(len(df) - 1) / (len(df) - (df.shape[1] - 1) - 1)
    return round(adj_rsqrd, 3)
```

[117]:
```python
lr=LinearRegression()
```

[118]:
```
lr.fit(x_train,y_train)
```

[118]:
```
LinearRegression()
```

[119]:
```
y_test=test_df['hc_mortgage_mean']
```

[120]:
```
y_test
```

[120]:
```
0       1139.24548
1       1533.25988
2       1254.54462
3        862.65763
4       1996.41425
           …
8931    1265.32007
8932    1079.67948
8933    1397.54610
8934    2890.43941
8935     872.73042
Name: hc_mortgage_mean, Length: 8673, dtype: float64
```

[121]:
```
x_test=test_df.drop(columns=['hc_mortgage_mean'])
```

[122]:
```
x_test
```

[122]:
```
          ALand   AWater   pop  male_pop  female_pop   rent_mean  rent_median  \
0       2711280    39555  3417      1479        1938   858.57169        859.0
1      14778785  2705204  3796      1846        1950   832.68625        750.0
2     258903666   863840  3944      2065        1879   816.00639        755.0
3     501694825  2623067  2508      1427        1081   418.68937        385.0
4      13796057   497689  6230      3274        2956  1031.63763        997.0
…           …       …     …        …           …           …
8931    2925514        0  4485      1849        2636   963.90313       1015.0
8932    9297182        0  4085      2136        1949   878.67414        849.0
8933    2699998        0  2891      1054        1837   706.77098        740.0
8934    4388948        0  4224      2142        2082  1616.74426       1759.0
8935  118940715  1641704  4627      2421        2206   597.34540        502.0

      rent_stdev  rent_sample_weight  rent_samples  …  female_age_mean  \
0      232.39082           276.07497         424.0  …         34.78682
1      267.22342           183.32299         245.0  …         44.23451
2      416.25699           141.39063         217.0  …         41.62426
3      156.92024            88.95960          93.0  …         44.81200
4      326.76727           277.39844         624.0  …         40.66618
…            …                  …           …  …            …
8931   425.25969           424.10678         828.0  …         33.60958
8932   344.79167           384.15938         620.0  …         33.37089
```

```
8933   380.02978           767.97204         1025.0  …         52.29779
8934   638.75073            75.30821          241.0  …         41.95954
8935   255.28214           287.58403          334.0  …         43.79431

      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
…                   …               …                       …
8931           31.16667          22.47299                 646.67694
8932           28.16667          22.18007                 475.14449
8933           51.58333          22.97228                 454.08417
8934           45.00000          22.28116                 503.69959
8935           45.83333          23.57733                 501.92109

      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
…                    …       …        …          …          …         …
8931              2636.0  0.38884  0.34566      0.10920    0.07137   0.17111
8932              1949.0  0.49400  0.39796      0.07443    0.02161   0.11765
8933              1837.0  0.37422  0.50055      0.05083    0.01215   0.11713
8934              2082.0  0.82867  0.54377      0.06435    0.02319   0.01681
8935              2206.0  0.79893  0.47189      0.04642    0.02630   0.14492

[8673 rows x 64 columns]
```

[123]:
```python
predict_test = lr.predict(x_test)
```

[124]:
```python
mae = mean_absolute_error(y_test, predict_test)
mse = mean_squared_error(y_test, predict_test)
r2 = r2_score(y_test, predict_test)

print("The model performance for test set")
print("--------------------------------------")
print('MAE is {}'.format(round(mae, 3)))
print('MSE is {}'.format(round(mse, 3)))
print('RMSE is {}'.format(round(mse**(0.5), 3)))
print('R2 score is {}'.format(round(r2, 3)))

print('Adjusted R2 score is {}'.format(adj_rsqrd(x_test, r2)))
```

```
The model performance for test set
--------------------------------------
MAE is 43.643
MSE is 4696.72
RMSE is 68.533
R2 score is 0.988
Adjusted R2 score is 0.988
```

[125]:
```
correlated_features = set()
correlation_matrix = train_df.drop('hc_mortgage_mean', axis=1).corr()

for i in range(len(correlation_matrix.columns)):
    for j in range(i):
        if abs(correlation_matrix.iloc[i, j]) > 0.8:
            colname = correlation_matrix.columns[i]
            correlated_features.add(colname)
```

[126]:
```
correlated_features
```

[126]:
```
{'debt_cdf',
 'family_mean',
 'family_median',
 'family_sample_weight',
 'family_samples',
 'family_stdev',
 'female_age_mean',
 'female_age_median',
 'female_age_sample_weight',
 'female_age_samples',
 'female_pop',
 'hc_mean',
 'hc_median',
 'hc_mortgage_samples',
 'hc_sample_weight',
 'hi_median',
 'hi_samples',
 'hi_stdev',
 'home_equity_cdf',
 'hs_degree_female',
 'hs_degree_male',
 'male_age_median',
 'male_age_sample_weight',
 'male_age_samples',
 'male_pop',
 'rent_gt_25',
 'rent_gt_30',
 'rent_gt_35',
```

```
 'rent_gt_40',
 'rent_gt_50',
 'rent_median',
 'rent_samples',
 'second_mortgage',
 'universe_samples',
 'used_samples'}
```

[127]:
```
corr_list = ['debt_cdf', 'family_mean', 'family_median',␣
 ↪'family_sample_weight', 'family_samples',
            'family_stdev', 'female_age_mean',␣
 ↪'female_age_median','female_age_sample_weight',
            'female_age_samples', 'female_pop', 'hc_median',␣
 ↪'hc_mortgage_samples', 'hc_sample_weight',
            'hi_median','hi_samples', 'hi_stdev', 'home_equity_cdf',␣
 ↪'hs_degree_female',
            'hs_degree_male', 'male_age_median',␣
 ↪'male_age_sample_weight','male_age_samples',
            'male_pop', 'rent_gt_25', 'rent_gt_30', 'rent_gt_35',␣
 ↪'rent_gt_40', 'rent_gt_50',
            'rent_median', 'rent_samples', 'second_mortgage',␣
 ↪'universe_samples', 'used_samples']
```

[128]:
```
train_df.drop(corr_list,axis=1,inplace=True)
```

[129]:
```
test_df.drop(corr_list,axis=1,inplace=True)
```

[130]:
```
train_df.shape
```

[130]:
```
(12165, 31)
```

[131]:
```
train_df.columns
```

[131]:
```
Index(['ALand', 'AWater', 'pop', 'rent_mean', 'rent_stdev',
       'rent_sample_weight', 'rent_gt_10', 'rent_gt_15', 'rent_gt_20',
       'hi_mean', 'hi_sample_weight', 'hc_mortgage_mean', 'hc_mortgage_median',
       'hc_mortgage_stdev', 'hc_mortgage_sample_weight', 'hc_mean', 'hc_stdev',
       'hc_samples', 'home_equity_second_mortgage', 'home_equity', 'debt',
       'second_mortgage_cdf', 'hs_degree', 'male_age_mean', 'male_age_stdev',
       'female_age_stdev', 'pct_own', 'married', 'married_snp', 'separated',
       'divorced'],
      dtype='object')
```

[132]:
```
test_df.shape
```

[132]:
```
(8673, 31)
```

[133]:
```python
test_df.columns
```

[133]:
```
Index(['ALand', 'AWater', 'pop', 'rent_mean', 'rent_stdev',
       'rent_sample_weight', 'rent_gt_10', 'rent_gt_15', 'rent_gt_20',
       'hi_mean', 'hi_sample_weight', 'hc_mortgage_mean', 'hc_mortgage_median',
       'hc_mortgage_stdev', 'hc_mortgage_sample_weight', 'hc_mean', 'hc_stdev',
       'hc_samples', 'home_equity_second_mortgage', 'home_equity', 'debt',
       'second_mortgage_cdf', 'hs_degree', 'male_age_mean', 'male_age_stdev',
       'female_age_stdev', 'pct_own', 'married', 'married_snp', 'separated',
       'divorced'],
      dtype='object')
```

[134]:
```python
X_train_1 = train_df.drop(columns=['hc_mortgage_mean'])
```

[135]:
```python
X_train_1.shape
```

[135]:
```
(12165, 30)
```

[136]:
```python
y_train_1 = train_df['hc_mortgage_mean']
```

[137]:
```python
y_train_1.shape
```

[137]:
```
(12165,)
```

[138]:
```python
X_test_1 = test_df.drop(columns=['hc_mortgage_mean'])
```

[139]:
```python
X_test_1.shape
```

[139]:
```
(8673, 30)
```

[140]:
```python
y_test_1 = test_df['hc_mortgage_mean']
```

[141]:
```python
y_test_1.shape
```

[141]:
```
(8673,)
```

[143]:
```python
lr.fit(X_train_1, y_train_1)
```

[143]:
```
LinearRegression()
```

[144]:
```python
predict_test_1 = lr.predict(X_test_1)
```

[145]:
```python
# model evaluation for testing set

mae = mean_absolute_error(y_test_1, predict_test_1)
mse = mean_squared_error(y_test_1, predict_test_1)
r2 = r2_score(y_test_1, predict_test_1)
```

```python
print("The model performance for test set")
print("--------------------------------------")
print('MAE is {}'.format(round(mae, 3)))
print('MSE is {}'.format(round(mse, 3)))
print('RMSE is {}'.format(round(mse**(0.5), 3)))
print('R2 score is {}'.format(round(r2, 3)))

print('Adjusted R2 score is {}'.format(adj_rsqrd(X_test_1, r2)))
```

```
The model performance for test set
--------------------------------------
MAE is 43.927
MSE is 4803.643
RMSE is 69.308
R2 score is 0.988
Adjusted R2 score is 0.988
```

[146]:
```python
sorted(SCORERS.keys())
```

[146]:
```
['accuracy',
 'adjusted_mutual_info_score',
 'adjusted_rand_score',
 'average_precision',
 'balanced_accuracy',
 'completeness_score',
 'explained_variance',
 'f1',
 'f1_macro',
 'f1_micro',
 'f1_samples',
 'f1_weighted',
 'fowlkes_mallows_score',
 'homogeneity_score',
 'jaccard',
 'jaccard_macro',
 'jaccard_micro',
 'jaccard_samples',
 'jaccard_weighted',
 'matthews_corrcoef',
 'max_error',
 'mutual_info_score',
 'neg_brier_score',
 'neg_log_loss',
 'neg_mean_absolute_error',
 'neg_mean_absolute_percentage_error',
 'neg_mean_gamma_deviance',
```

```
 'neg_mean_poisson_deviance',
 'neg_mean_squared_error',
 'neg_mean_squared_log_error',
 'neg_median_absolute_error',
 'neg_negative_likelihood_ratio',
 'neg_root_mean_squared_error',
 'normalized_mutual_info_score',
 'positive_likelihood_ratio',
 'precision',
 'precision_macro',
 'precision_micro',
 'precision_samples',
 'precision_weighted',
 'r2',
 'rand_score',
 'recall',
 'recall_macro',
 'recall_micro',
 'recall_samples',
 'recall_weighted',
 'roc_auc',
 'roc_auc_ovo',
 'roc_auc_ovo_weighted',
 'roc_auc_ovr',
 'roc_auc_ovr_weighted',
 'top_k_accuracy',
 'v_measure_score']
```

[147]:
```
import random
randomlist = []
for i in range(0,100):
    n = random.randint(1,len(X_test_1))
    randomlist.append(n)
print(randomlist)
```

```
[6310, 6291, 3255, 6815, 6451, 1612, 5577, 23, 6869, 6374, 3869, 7717, 4785,
5969, 5345, 6098, 4116, 3626, 8317, 7471, 4711, 7774, 6678, 907, 8146, 5756,
4485, 2663, 4339, 8448, 6151, 7719, 602, 1001, 8659, 2621, 3937, 3858, 8204,
4083, 6324, 1065, 2677, 6035, 6052, 2207, 198, 2216, 1136, 2572, 5719, 2629,
2499, 593, 8113, 7248, 5623, 5320, 7375, 3037, 6472, 6486, 6336, 4544, 3397,
4227, 4139, 3544, 2259, 1736, 7758, 5158, 2381, 4492, 2171, 1403, 8238, 146,
6375, 179, 226, 5203, 3857, 4669, 2719, 5928, 6769, 4782, 4810, 7771, 2409,
5738, 3650, 6940, 3710, 6582, 6335, 927, 6694, 2684]
```

[148]:
```
pre_out = []
out = []
```

```python
for i in randomlist:
    data_in = [list(X_test_1.iloc[i])]
    pre_data_out = lr.predict(data_in)
    data_out = y_test_1.iloc[i]

    print(i, pre_data_out, data_out)

    pre_out.append(pre_data_out)
    out.append(data_out)
```

```
6310 [1730.2909608] 1638.26813
6291 [1411.24938018] 1391.98538
3255 [1255.19950288] 1235.90965
6815 [1061.2895463] 1081.70852
6451 [2296.68839923] 2387.85517
1612 [1529.22333263] 1560.47016
5577 [1561.69845022] 1561.20162
23 [1297.47504432] 1281.74697
6869 [2041.418827] 2010.07472
6374 [3722.46052632] 3486.71528
3869 [1053.34208531] 1036.89814
7717 [1024.94881752] 992.91347
4785 [1325.99223476] 1351.26489
5969 [842.15936646] 855.37876
5345 [2116.3224463] 2129.14226
6098 [1761.0863719] 1724.335
4116 [2450.40267694] 2439.81684
3626 [1415.72430636] 1402.49141
8317 [2769.92365661] 2845.118
7471 [1710.30601542] 1683.16974
4711 [1895.28228647] 1884.18001
7774 [1392.06703072] 1395.00406
6678 [1594.63001036] 1589.89502
907 [2523.17240713] 2552.8794
8146 [863.74498417] 836.91412
5756 [1123.26455413] 1134.64681
4485 [3233.42639279] 3230.56761
2663 [1414.83796912] 1365.77125
4339 [2051.01755862] 2063.31606
8448 [1523.72876149] 1462.89576
6151 [1297.12877891] 1261.28117
7719 [1243.24924774] 1249.36037
602 [1249.74286298] 1193.85512
1001 [2643.79276115] 2737.91429
8659 [904.10330899] 898.93585
2621 [1210.32964023] 1210.26883
3937 [1080.51355544] 1086.35404
```

3858 [1276.51341253] 1222.13931
8204 [1445.98177823] 1414.91406
4083 [1284.27087092] 1247.88416
6324 [958.53724672] 950.92537
1065 [1155.13871062] 1098.64518
2677 [1185.77634408] 1274.0943
6035 [967.90372384] 953.69531
6052 [1290.06762202] 1287.04525
2207 [1689.76489491] 1671.80112
198 [1390.21358428] 1357.1561
2216 [1053.62095338] 1099.66705
1136 [930.64052138] 945.19373
2572 [2094.19650767] 2098.0648
5719 [940.98682106] 915.52638
2629 [2347.15132426] 2318.79992
2499 [2243.38984091] 2224.44526
593 [1202.67016726] 1144.08759
8113 [1816.48469273] 1783.30068
7248 [2572.31654496] 2588.83154
5623 [3394.17938654] 3358.6395
5320 [1237.87455786] 1185.16754
7375 [1921.65759257] 1943.04375
3037 [2297.60808822] 2446.32575
6472 [1048.4413265] 1063.39641
6486 [1930.91237992] 2006.59914
6336 [973.06996026] 965.28054
4544 [1908.45834909] 1940.97732
3397 [1390.22813691] 1411.06856
4227 [1001.78267771] 996.66084
4139 [1734.8788838] 1759.28317
3544 [1120.41760312] 1150.53233
2259 [1718.40068301] 1777.50455
1736 [983.65834557] 1016.80588
7758 [1629.72266359] 1783.09229
5158 [582.44288633] 559.72369
2381 [1844.63356969] 1862.56939
4492 [2356.26262701] 2382.20749
2171 [2312.43268777] 2445.81826
1403 [617.42010577] 649.5
8238 [3254.61395232] 3136.4521
146 [896.53182975] 994.17727
6375 [1650.67250274] 1641.33822
179 [1253.17500219] 1245.58043
226 [2102.12135041] 2104.26409
5203 [2177.01216434] 2186.09115
3857 [1771.17159001] 1795.07134
4669 [1282.46174156] 1262.41042
2719 [1615.11832287] 1597.74797

```
5928 [1399.6831075] 1340.94815
6769 [1557.14183688] 1675.53328
4782 [3028.10417469] 3192.36072
4810 [1915.27816907] 1935.94698
7771 [811.26103133] 799.9851
2409 [3146.74329319] 3173.02499
5738 [2313.5728209] 2496.82892
3650 [1925.40810249] 1932.83543
6940 [2151.99653352] 2108.13743
3710 [1918.2996048] 1933.17841
6582 [1076.24242525] 1082.42413
6335 [1812.02631508] 1837.18698
927 [1301.43424513] 1297.71458
6694 [1409.77150299] 1404.56251
2684 [2536.82163811] 2531.9337
```

[149]:
```
pre_out
```

[149]:
```
[array([1730.2909608]),
 array([1411.24938018]),
 array([1255.19950288]),
 array([1061.2895463]),
 array([2296.68839923]),
 array([1529.22333263]),
 array([1561.69845022]),
 array([1297.47504432]),
 array([2041.418827]),
 array([3722.46052632]),
 array([1053.34208531]),
 array([1024.94881752]),
 array([1325.99223476]),
 array([842.15936646]),
 array([2116.3224463]),
 array([1761.0863719]),
 array([2450.40267694]),
 array([1415.72430636]),
 array([2769.92365661]),
 array([1710.30601542]),
 array([1895.28228647]),
 array([1392.06703072]),
 array([1594.63001036]),
 array([2523.17240713]),
 array([863.74498417]),
 array([1123.26455413]),
 array([3233.42639279]),
 array([1414.83796912]),
 array([2051.01755862]),
```

```
array([1523.72876149]),
array([1297.12877891]),
array([1243.24924774]),
array([1249.74286298]),
array([2643.79276115]),
array([904.10330899]),
array([1210.32964023]),
array([1080.51355544]),
array([1276.51341253]),
array([1445.98177823]),
array([1284.27087092]),
array([958.53724672]),
array([1155.13871062]),
array([1185.77634408]),
array([967.90372384]),
array([1290.06762202]),
array([1689.76489491]),
array([1390.21358428]),
array([1053.62095338]),
array([930.64052138]),
array([2094.19650767]),
array([940.98682106]),
array([2347.15132426]),
array([2243.38984091]),
array([1202.67016726]),
array([1816.48469273]),
array([2572.31654496]),
array([3394.17938654]),
array([1237.87455786]),
array([1921.65759257]),
array([2297.60808822]),
array([1048.4413265]),
array([1930.91237992]),
array([973.06996026]),
array([1908.45834909]),
array([1390.22813691]),
array([1001.78267771]),
array([1734.8788838]),
array([1120.41760312]),
array([1718.40068301]),
array([983.65834557]),
array([1629.72266359]),
array([582.44288633]),
array([1844.63356969]),
array([2356.26262701]),
array([2312.43268777]),
array([617.42010577]),
```

```
 array([3254.61395232]),
 array([896.53182975]),
 array([1650.67250274]),
 array([1253.17500219]),
 array([2102.12135041]),
 array([2177.01216434]),
 array([1771.17159001]),
 array([1282.46174156]),
 array([1615.11832287]),
 array([1399.6831075]),
 array([1557.14183688]),
 array([3028.10417469]),
 array([1915.27816907]),
 array([811.26103133]),
 array([3146.74329319]),
 array([2313.5728209]),
 array([1925.40810249]),
 array([2151.99653352]),
 array([1918.2996048]),
 array([1076.24242525]),
 array([1812.02631508]),
 array([1301.43424513]),
 array([1409.77150299]),
 array([2536.82163811])]
```

[150]:

```python
x = [2,3,5,9,1,0,2,3]

def my_min(sequence):
    """return the minimum element of sequence"""
    low = sequence[0] # need to start with some value
    for i in sequence:
        if i < low:
            low = i
    return low

print(my_min(x))
```

```
0
```

[151]:

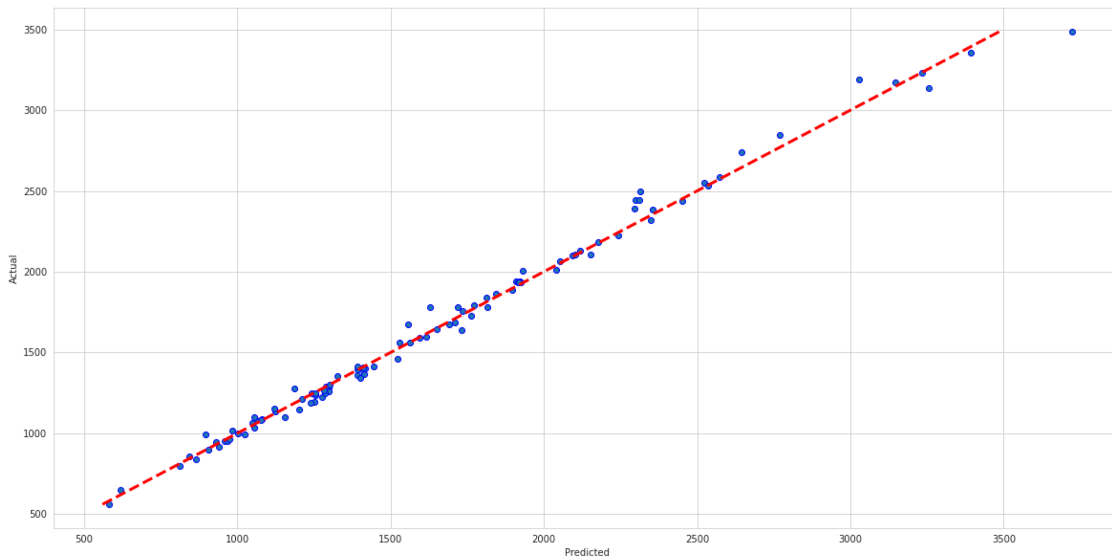
```python
x = [2,3,5,9,1,0,2,3]

def my_maxi(sequence):
    """return the minimum element of sequence"""
    maxi = sequence[0] # need to start with some value
    for i in sequence:
        if i > maxi:
            maxi = i
```

```
    return maxi

print(my_maxi(x))
```

9

[152]:

```
fig, ax = plt.subplots(figsize=(20,10))
ax.scatter(pre_out, out, edgecolors=(0, 0, 1))
ax.plot([my_min(out), my_maxi(out)], [my_min(out), my_maxi(out)], 'r--', lw=3)
ax.set_xlabel('Predicted')
ax.set_ylabel('Actual')
plt.show()
```



[153]:

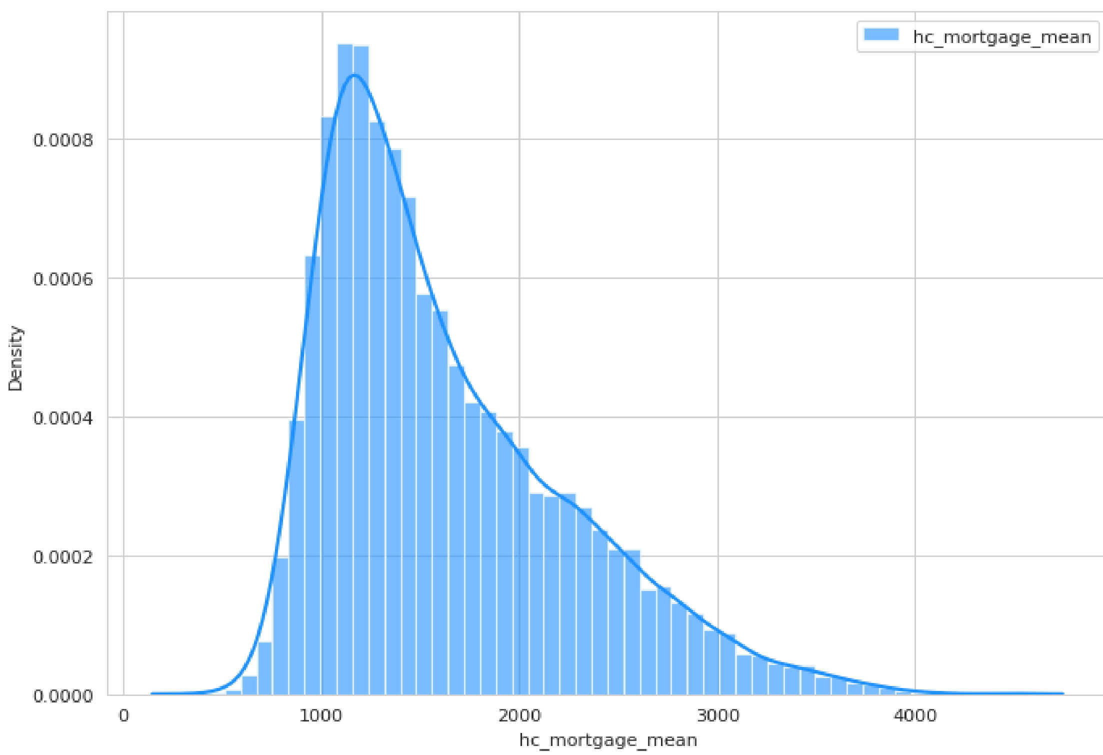
```
# model evaluation for testing set

mae = mean_absolute_error(y_test_1, predict_test_1)
mse = mean_squared_error(y_test_1, predict_test_1)
r2 = r2_score(y_test_1, predict_test_1)

print("The model performance for test set")
print("--------------------------------------")
print('MAE is {}'.format(round(mae, 3)))
print('MSE is {}'.format(round(mse, 3)))
print('RMSE is {}'.format(round(mse**(0.5), 3)))
print('R2 score is {}'.format(round(r2, 3)))

print('Adjusted R2 score is {}'.format(adj_rsqrd(X_test_1, r2)))
```

The model performance for test set

```
--------------------------------------
MAE is 43.927
MSE is 4803.643
RMSE is 69.308
R2 score is 0.988
Adjusted R2 score is 0.988
```

[154]:

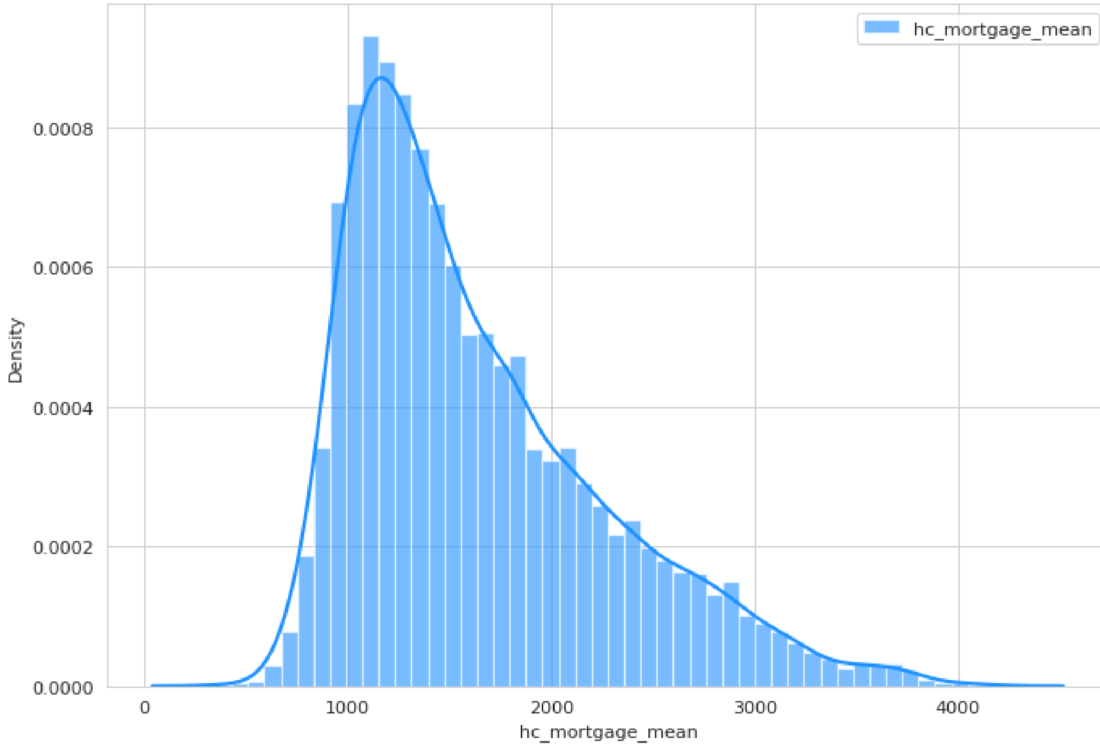
```
# Plot
kwargs = dict(hist_kws={'alpha':.6}, kde_kws={'linewidth':2})

plt.figure(figsize=(10,7), dpi= 80)
sns.distplot(y_train_1, color="dodgerblue", label="hc_mortgage_mean", **kwargs)
# sns.distplot(x2, color="orange", label="SUV", **kwargs)
# sns.distplot(x3, color="deeppink", label="minivan", **kwargs)
# plt.xlim(50,75)
plt.legend();
```



[155]:

```
# Plot
kwargs = dict(hist_kws={'alpha':.6}, kde_kws={'linewidth':2})

plt.figure(figsize=(10,7), dpi= 80)
sns.distplot(y_test_1, color="dodgerblue", label="hc_mortgage_mean", **kwargs)
# sns.distplot(x2, color="orange", label="SUV", **kwargs)
```

```python
# sns.distplot(x3, color="deeppink", label="minivan", **kwargs)
# plt.xlim(50,75)
plt.legend();
```



0.0.10 11. Create a dashboard in tableau by choosing appropriate chart types and metrics useful for the business. The dashboard must entail the following:

a) Box plot of distribution of average rent by type of place (Village, urban, town etc.)
b) Pie charts (Venn diagram) to show overall debt (% bad and good debt) and bad debt (2 mortgage and home equity loan)
c) Explore the top 2,500 locations where the percentage of households with a second mortgage is the highest and percent ownership is above 10%. Visualize using geo-map.
d) Heat map for correlation matrix
e) Pie chart to show the population distribution across different types of places (Village, urban, town etc.)

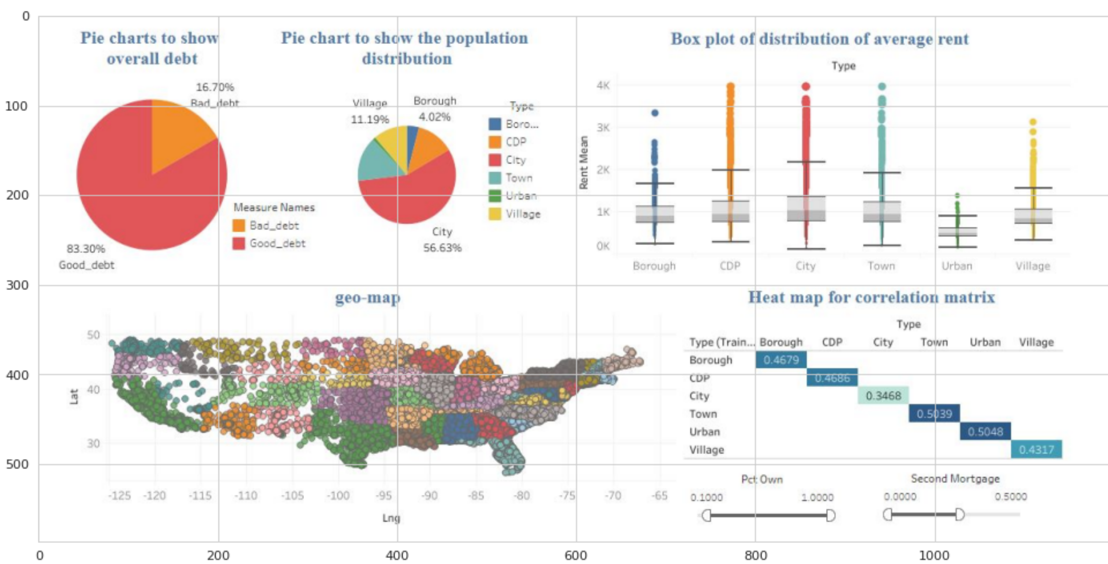
https://public.tableau.com/app/profile/shubhangi.yerkal/viz/RealEstate_16914987986460/Dashboard1?publish=

[156]:
```python
from PIL import Image
```

[157]:
```python
img=Image.open('Tableau screenshot.JPG')
```

```
[158]: plt.figure(figsize=(15,10),dpi=80)
       plt.imshow(img)
```

```
[158]: <matplotlib.image.AxesImage at 0x7fb4944dc100>
```



0.0.11 9. 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 depend on 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 1. Highschool graduation rates 2. Median population age 3. Second Mortgage Statistics 4. Percent Own 5. Bad Debt Expense

```
[159]: train_new_df=pd.read_csv('train.csv')
```

```
[160]: test_new_df=pd.read_csv('test.csv')
```

```
[161]: train_new_df.shape
```

```
[161]: (12513, 80)
```

```
[162]: test_new_df.shape
```

```
[162]: (8937, 80)
```

[163]:
```
def cat_variables(df):
    cat_variables = list(df.select_dtypes(exclude = ['int', 'float']).columns)
    return cat_variables
```

[164]:
```
def num_variables(df):
    num_variables = list(df.select_dtypes(include = ['int', 'float']).columns)
    return num_variables
```

[165]:
```
cat_variables(train_new_df)
```

[165]:
```
['state', 'state_ab', 'city', 'place', 'type', 'primary', 'lng']
```

[166]:
```
num_variables(train_new_df)
```

[166]:
```
['UID',
 'BLOCKID',
 'SUMLEVEL',
 'COUNTYID',
 'STATEID',
 'zip_code',
 'area_code',
 'lat',
 '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']
```

[167]:
```
fa_train_df = train_new_df[num_variables(train_new_df)]
fa_train_df
```

[167]:
```
      UID  BLOCKID  SUMLEVEL  COUNTYID  STATEID  zip_code  area_code  \
0  267822      NaN       140        53       36     13346        315
1  246444      NaN       140       141       18     46616        574
```

```
2      245683      NaN       140        63        18     46122         317
3      279653      NaN       140       127        72       927         787
4      247218      NaN       140       161        20     66502         785
…        …      …       …        …        …        …        …
12508  255188      NaN       140       161        26     48197         734
12509  240397      NaN       140       245        13     30906         706
12510  251120      NaN       140        27        25      1420         978
12511  291183      NaN       140        33        53     98117         206
12512  256921      NaN       140       137        27     55803         218

             lat        ALand      AWater  …  female_age_mean  \
0      42.840812  202183361.0  1699120.0  …         44.48629
1      41.701441    1560828.0   100363.0  …         36.48391
2      39.792202   69561595.0   284193.0  …         42.15810
3      18.396103    1105793.0        0.0  …         47.77526
4      39.195573    2554403.0        0.0  …         24.17693
…            …            …          …  …               …
12508  42.220559    1892447.0     2004.0  …         30.87275
12509  33.434136    6742997.0    27745.0  …         43.45140
12510  42.576436    3211646.0    10033.0  …         37.82698
12511  47.685096    1493249.0        0.0  …         38.97593
12512  46.853471          NaN        NaN  …              NaN

       female_age_median  female_age_stdev  female_age_sample_weight  \
0               45.33333          22.51276                 685.33845
1               37.58333          23.43353                 267.23367
2               42.83333          23.94119                 707.01963
3               50.58333          24.32015                 362.20193
4               21.58333          11.10484                1854.48652
…                    …                 …                        …
12508           25.91667          15.69777                 315.07146
12509           46.16667          23.73648                 804.60756
12510           33.25000          24.45782                 764.81692
12511           38.83333          20.66153                 704.81384
12512                NaN               NaN                       NaN

       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
…                     …        …       …            …         …
12508              1029.0  0.08881  0.29096      0.01835    0.00655   0.08257
12509              3062.0  0.50141  0.41594      0.05894    0.02995   0.09903
12510              3179.0  0.58918  0.44332      0.04636    0.02610   0.09272
12511              2868.0  0.67227  0.53217      0.00046    0.00000   0.10524
```

```
12512                    NaN       NaN       NaN          NaN         NaN        NaN

[12513 rows x 73 columns]
```

[168]:
```
# exclude columns you don't want
fa_train_df = fa_train_df[fa_train_df.columns[~fa_train_df.columns.
 ↪isin(['SUMLEVEL', 'lat', 'lng', 'ALand', 'AWater'])]]
```

[169]:
```
from factor_analyzer import FactorAnalyzer
import warnings
warnings.filterwarnings('ignore')
```

[170]:
```
fa_train_df .shape
```

[170]:
```
(12513, 69)
```

[171]:
```
fa_train_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12513 entries, 0 to 12512
Data columns (total 69 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   UID                           12513 non-null  int64
 1   BLOCKID                       0 non-null      float64
 2   COUNTYID                      12513 non-null  int64
 3   STATEID                       12513 non-null  int64
 4   zip_code                      12513 non-null  int64
 5   area_code                     12513 non-null  int64
 6   pop                           12512 non-null  float64
 7   male_pop                      12512 non-null  float64
 8   female_pop                    12512 non-null  float64
 9   rent_mean                     12372 non-null  float64
 10  rent_median                   12372 non-null  float64
 11  rent_stdev                    12372 non-null  float64
 12  rent_sample_weight            12372 non-null  float64
 13  rent_samples                  12372 non-null  float64
 14  rent_gt_10                    12372 non-null  float64
 15  rent_gt_15                    12372 non-null  float64
 16  rent_gt_20                    12372 non-null  float64
 17  rent_gt_25                    12372 non-null  float64
 18  rent_gt_30                    12372 non-null  float64
 19  rent_gt_35                    12372 non-null  float64
 20  rent_gt_40                    12372 non-null  float64
 21  rent_gt_50                    12372 non-null  float64
 22  universe_samples              12512 non-null  float64
 23  used_samples                  12512 non-null  float64
```

```
 24  hi_mean                      12388 non-null  float64
 25  hi_median                    12388 non-null  float64
 26  hi_stdev                     12388 non-null  float64
 27  hi_sample_weight             12388 non-null  float64
 28  hi_samples                   12388 non-null  float64
 29  family_mean                  12372 non-null  float64
 30  family_median                12372 non-null  float64
 31  family_stdev                 12372 non-null  float64
 32  family_sample_weight         12372 non-null  float64
 33  family_samples               12372 non-null  float64
 34  hc_mortgage_mean             12247 non-null  float64
 35  hc_mortgage_median           12247 non-null  float64
 36  hc_mortgage_stdev            12247 non-null  float64
 37  hc_mortgage_sample_weight    12247 non-null  float64
 38  hc_mortgage_samples          12247 non-null  float64
 39  hc_mean                      12222 non-null  float64
 40  hc_median                    12222 non-null  float64
 41  hc_stdev                     12222 non-null  float64
 42  hc_samples                   12222 non-null  float64
 43  hc_sample_weight             12222 non-null  float64
 44  home_equity_second_mortgage  12297 non-null  float64
 45  second_mortgage              12297 non-null  float64
 46  home_equity                  12297 non-null  float64
 47  debt                         12297 non-null  float64
 48  second_mortgage_cdf          12297 non-null  float64
 49  home_equity_cdf              12297 non-null  float64
 50  debt_cdf                     12297 non-null  float64
 51  hs_degree                    12422 non-null  float64
 52  hs_degree_male               12420 non-null  float64
 53  hs_degree_female             12404 non-null  float64
 54  male_age_mean                12423 non-null  float64
 55  male_age_median              12423 non-null  float64
 56  male_age_stdev               12423 non-null  float64
 57  male_age_sample_weight       12423 non-null  float64
 58  male_age_samples             12423 non-null  float64
 59  female_age_mean              12412 non-null  float64
 60  female_age_median            12412 non-null  float64
 61  female_age_stdev             12412 non-null  float64
 62  female_age_sample_weight     12412 non-null  float64
 63  female_age_samples           12412 non-null  float64
 64  pct_own                      12388 non-null  float64
 65  married                      12422 non-null  float64
 66  married_snp                  12422 non-null  float64
 67  separated                    12422 non-null  float64
 68  divorced                     12422 non-null  float64
dtypes: float64(64), int64(5)
memory usage: 6.6 MB
```

[172]:
```python
fa_train_df.drop(columns=['BLOCKID'],axis=1,inplace=True)
```

[173]:
```python
fa_train_df.shape
```

[173]:
```
(12513, 68)
```

[174]:
```python
len(fa_train_df.columns[fa_train_df.isnull().sum(axis=0)>0])
```

[174]:
```
63
```

[175]:
```python
null_rows_2=fa_train_df[fa_train_df.isnull().any(axis=1)]
null_rows_2
```

[175]:
```
            UID  COUNTYID  STATEID  zip_code  area_code     pop  male_pop  \
51       223593        19        4     85734        520  4531.0    4370.0
94       233040       101        8     81001        719   579.0     270.0
153      263292        13       34      7107        973  3458.0    1787.0
302      267158        47       36     11215        718     0.0       0.0
340      292484        25       55     53703        608  3274.0    1293.0
…        …         …        …      …          …      …       …
12338    279610       127       72       928        787  2266.0     834.0
12361    274458       109       40     73102        405   182.0     115.0
12435    290374       710       51     23502        757     0.0       0.0
12494    246025        95       18     46060        765  3518.0    3509.0
12512    256921       137       27     55803        218     NaN       NaN

       female_pop   rent_mean  rent_median  …  female_age_mean  \
51          161.0         NaN          NaN  …         40.02370
94          309.0   782.00000        781.0  …         20.00784
153        1671.0   890.69365        929.0  …         35.47667
302           0.0         NaN          NaN  …              NaN
340        1981.0  1191.78679        956.0  …         22.03226
…             …          …         …  …             …
12338      1432.0   147.54810        104.0  …         26.77626
12361        67.0   283.80307        220.0  …         59.38249
12435         0.0         NaN          NaN  …              NaN
12494         9.0   646.12963        645.0  …         54.28123
12512         NaN         NaN          NaN  …              NaN

       female_age_median  female_age_stdev  female_age_sample_weight  \
51              40.83333           8.49563                  30.01695
94              19.25000           4.30291                 172.56153
153             35.58333          20.62717                 369.61740
302                  NaN               NaN                       NaN
340             21.08333           5.13435                1365.86300
…                   …               …                       …
12338           24.41667          19.03316                 366.92156
```

```
12361          64.16667       13.96468                 20.66249
12435               NaN            NaN                      NaN
12494          54.25000        2.78274                  1.67797
12512               NaN            NaN                      NaN

       female_age_samples  pct_own  married  married_snp  separated  divorced
51                  161.0      NaN  0.16308      0.16308    0.02634   0.20499
94                  309.0  0.00000  0.00000      0.00000    0.00000   0.00000
153                1671.0  0.24002  0.37411      0.05579    0.02504   0.07654
302                   NaN      NaN      NaN          NaN        NaN       NaN
340                1981.0  0.00000  0.00773      0.00000    0.00000   0.01160
…                     …      …      …          …        …       …
12338              1432.0  0.00000  0.03865      0.00000    0.00000   0.05314
12361                67.0  0.02198  0.11712      0.04505    0.00000   0.48649
12435                 NaN      NaN      NaN          NaN        NaN       NaN
12494                 9.0  0.00000  0.10288      0.10288    0.02337   0.25677
12512                 NaN      NaN      NaN          NaN        NaN       NaN

[348 rows x 68 columns]
```

[176]:
```
(348/12513)*100
```

[176]: 2.781107648046032

[177]:
```
fa_train_df = pd.concat([fa_train_df, null_rows_2, null_rows_2]).
 ↪drop_duplicates(keep=False)
```

[178]:
```
fa_train_df.shape
```

[178]: (12165, 68)

[179]:
```
len(fa_train_df.columns[fa_train_df.isnull().sum(axis=0)>0])
```

[179]: 0

[180]:
```
# Create factor analysis object and perform factor analysis
fa = FactorAnalyzer( rotation=None, n_factors = 25)
```

[181]:
```
train_df.shape
```

[181]: (12165, 31)

[183]:
```
fa.fit(train_df)
```

[183]: FactorAnalyzer(n_factors=25, rotation=None, rotation_kwargs={})

[184]:
```python
# Check Eigenvalues
ev, v = fa.get_eigenvalues()
ev
```

[184]:
```
array([7.57088436e+00, 4.27512392e+00, 2.99161913e+00, 2.14029845e+00,
       1.93924290e+00, 1.52638565e+00, 1.34953352e+00, 1.30398770e+00,
       9.72458739e-01, 8.15347435e-01, 6.79878139e-01, 6.33025373e-01,
       5.73236087e-01, 5.24915522e-01, 4.70358801e-01, 3.99501375e-01,
       3.75916007e-01, 3.50847623e-01, 3.11498251e-01, 2.83232555e-01,
       2.39088974e-01, 2.20751676e-01, 2.14026568e-01, 2.07564544e-01,
       1.55515677e-01, 1.44324704e-01, 1.14221722e-01, 1.00396412e-01,
       7.86851634e-02, 3.15268184e-02, 6.60621157e-03])
```

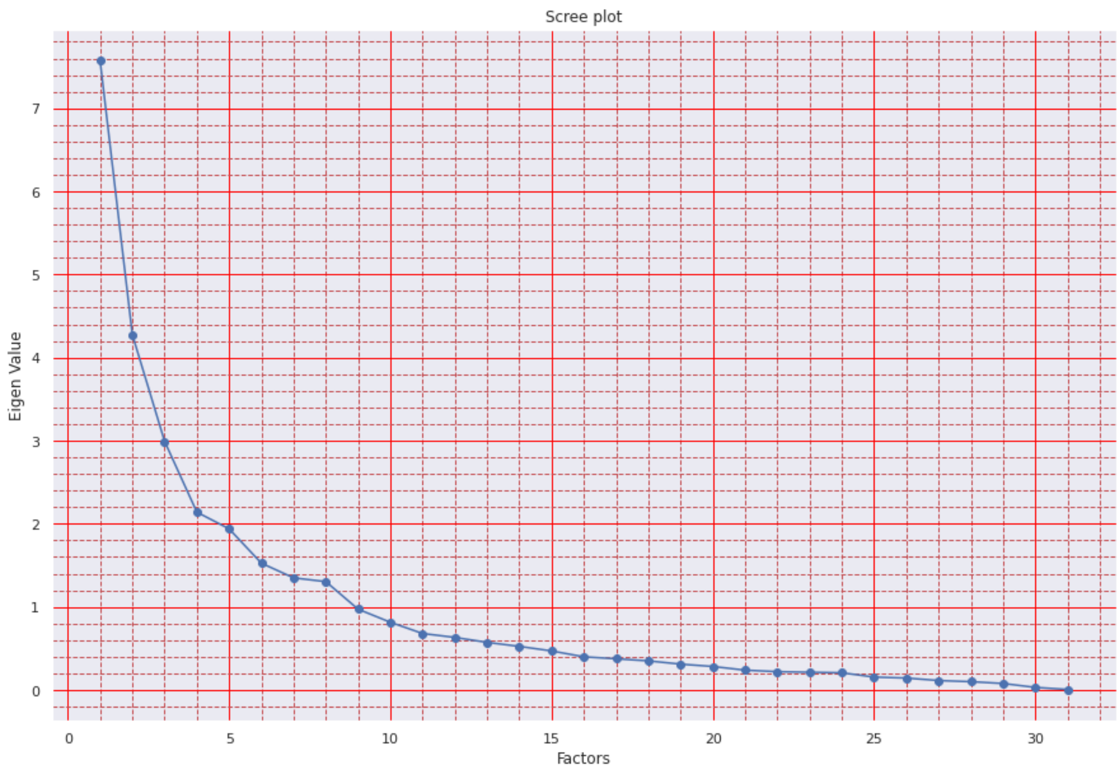
[185]:
```python
print(sorted(ev, reverse=True))
```

```
[7.570884355752323, 4.275123920017025, 2.9916191310847178, 2.1402984455089493,
1.9392429047237474, 1.526385648338118, 1.3495335157039268, 1.3039877008307375,
0.9724587386818055, 0.815347435430352, 0.6798781385988164, 0.6330253728442834,
0.5732360866167423, 0.5249155223088642, 0.4703588012181757, 0.39950137515259526,
0.37591600717833207, 0.3508476233627324, 0.3114982511873748, 0.2832325545961217,
0.23908897406525612, 0.22075167614486443, 0.21402656800676365,
0.20756454405390867, 0.1555156774633129, 0.1443247039725599,
0.11422172219152367, 0.10039641163846852, 0.07868516336278969,
0.03152681839220227, 0.006606211572608851]
```

[186]:
```python
loadings = fa.loadings_
```

[187]:
```python
xvals = range(1, train_df.shape[1]+1)
```

[188]:
```python
sns.set()
plt.figure(figsize = (15,10))
plt.scatter(xvals, ev)
plt.plot(xvals, ev)
plt.title('Scree plot')
plt.xlabel('Factors')
plt.ylabel('Eigen Value')
plt.grid(color = 'red', )
plt.grid(b=True, which='minor', color='r', linestyle='--')
plt.minorticks_on()
plt.show()
```



[189]:
```
Factors  = pd.DataFrame.from_records(loadings)

Factors = Factors.add_prefix('Factor ')

Factors.index = train_df.columns
Factors
```

[189]:

	Factor 0	Factor 1	Factor 2	Factor 3	Factor 4	\
ALand	-0.048828	0.090813	-0.113211	0.013413	-0.089836	
AWater	-0.011884	0.023831	-0.061075	0.011872	-0.051161	
pop	0.104358	0.172644	0.804188	0.391166	-0.142843	
rent_mean	0.824324	-0.184323	0.078809	0.079004	0.089530	
rent_stdev	0.663397	-0.144518	0.005719	0.223928	0.035128	
rent_sample_weight	-0.491344	-0.295950	0.348741	0.335033	-0.197240	
rent_gt_10	-0.011768	-0.300154	0.291500	-0.030387	0.565749	
rent_gt_15	-0.035590	-0.419071	0.291660	0.016761	0.714264	
rent_gt_20	-0.105444	-0.428885	0.195628	0.056253	0.613790	
hi_mean	0.924545	0.172582	0.020106	-0.003689	-0.074470	
hi_sample_weight	-0.399799	0.160713	0.722457	0.457978	-0.073261	
hc_mortgage_mean	0.904773	-0.236484	-0.037600	0.235390	-0.005160	
hc_mortgage_median	0.887771	-0.247313	-0.038491	0.229018	-0.007489	
hc_mortgage_stdev	0.776804	-0.052197	-0.069370	0.238895	0.022907	

hc_mortgage_sample_weight	-0.142982	0.591082	0.690958	0.010959	-0.044319
hc_mean	0.794804	-0.158146	-0.093336	0.286485	-0.020209
hc_stdev	0.622993	-0.045508	-0.135105	0.351689	0.005998
hc_samples	-0.089225	0.772069	0.259814	0.380170	0.095301
home_equity_second_mortgage	0.189348	-0.221390	0.406098	-0.547625	-0.040470
home_equity	0.595954	-0.128115	0.290038	-0.352058	0.002389
debt	0.496456	-0.311243	0.483726	-0.330157	-0.106966
second_mortgage_cdf	-0.285522	0.018456	-0.411368	0.506677	-0.009657
hs_degree	0.549801	0.313776	0.103469	-0.162389	-0.067352
male_age_mean	0.205078	0.516032	-0.275021	0.080718	0.254210
male_age_stdev	0.009809	0.605360	-0.159099	0.006569	0.383330
female_age_stdev	-0.102541	0.484053	-0.145496	0.029041	0.344212
pct_own	0.373209	0.756136	-0.010695	-0.230973	0.167345
married	0.458956	0.628252	0.030597	-0.062142	0.122659
married_snp	-0.291069	-0.415067	-0.097445	0.246585	0.064955
separated	-0.340516	-0.300866	-0.095115	0.174423	0.103135
divorced	-0.451219	0.085865	-0.060982	-0.075404	0.069670

	Factor 5	Factor 6	Factor 7	Factor 8	Factor 9 \
ALand	0.005027	0.051229	0.331032	0.440243	0.228499
AWater	-0.005891	0.032348	0.315456	0.406168	0.253254
pop	-0.031574	0.175452	-0.013508	0.098132	-0.077378
rent_mean	-0.106863	0.144683	0.016082	0.012907	0.059252
rent_stdev	0.070333	-0.042576	-0.036837	0.103815	0.059584
rent_sample_weight	0.219230	-0.358922	-0.147927	0.042800	0.079679
rent_gt_10	-0.208417	-0.021710	0.036211	-0.045843	0.040461
rent_gt_15	-0.191387	-0.045079	0.082552	0.027723	0.012719
rent_gt_20	-0.109786	-0.046611	0.071386	0.048186	-0.031833
hi_mean	-0.072259	0.101292	-0.045123	-0.018880	0.071197
hi_sample_weight	0.132644	-0.164815	0.017609	-0.007797	0.033621
hc_mortgage_mean	0.083150	-0.024151	-0.003989	0.026733	-0.046895
hc_mortgage_median	0.074962	-0.014640	-0.019970	0.037606	-0.044546
hc_mortgage_stdev	0.156015	-0.086726	0.091219	-0.015077	-0.039971
hc_mortgage_sample_weight	-0.109272	0.151467	-0.038131	-0.061264	0.150542
hc_mean	0.043620	-0.124703	-0.013342	-0.059974	0.004299
hc_stdev	0.118404	-0.132240	0.098370	-0.063383	-0.024078
hc_samples	0.018637	0.037317	0.272227	-0.094348	-0.162578
home_equity_second_mortgage	0.463315	-0.043094	0.239159	0.016186	-0.222251
home_equity	0.237296	-0.094405	-0.026536	-0.016556	0.049773
debt	-0.098835	0.099492	-0.314450	0.005676	0.317503
second_mortgage_cdf	-0.369992	0.034975	-0.158287	-0.008282	0.110162
hs_degree	-0.093992	-0.261313	0.042769	-0.162544	0.284748
male_age_mean	0.184478	-0.182156	0.208458	-0.261374	0.162243
male_age_stdev	0.326976	-0.086004	-0.303535	0.227829	-0.004862
female_age_stdev	0.377167	-0.104903	-0.329997	0.227039	0.013578
pct_own	-0.137556	0.220599	0.078474	-0.065005	0.006794
married	0.020504	0.278891	-0.094848	0.011126	-0.021239

married_snp	0.398917	0.493057	0.028806	-0.099693	0.110368
separated	0.410782	0.458642	0.021598	-0.191915	0.222886
divorced	0.160108	-0.321517	0.105472	-0.203901	0.303771

	…	Factor 15	Factor 16	Factor 17	Factor 18	\
ALand	…	-0.007849	0.002780	0.010913	-0.007802	
AWater	…	0.006339	0.020963	0.007846	-0.001478	
pop	…	0.020721	-0.051661	0.006628	0.106867	
rent_mean	…	-0.115969	-0.167152	0.152891	-0.102881	
rent_stdev	…	0.130698	0.161761	-0.058931	0.043745	
rent_sample_weight	…	-0.056341	-0.023716	-0.043537	0.084596	
rent_gt_10	…	0.159255	0.104221	0.002569	0.010122	
rent_gt_15	…	-0.050045	-0.059281	-0.045308	0.035190	
rent_gt_20	…	-0.086732	-0.043319	-0.007275	0.002313	
hi_mean	…	-0.007182	-0.119538	-0.101949	0.134942	
hi_sample_weight	…	-0.073055	0.017876	0.046690	-0.050745	
hc_mortgage_mean	…	-0.046299	0.064326	-0.056652	0.002272	
hc_mortgage_median	…	-0.064498	0.091460	-0.050350	0.024607	
hc_mortgage_stdev	…	0.043653	-0.081422	-0.063091	-0.076556	
hc_mortgage_sample_weight	…	0.039827	0.029910	0.008235	-0.076041	
hc_mean	…	-0.005504	0.103590	0.000178	-0.108906	
hc_stdev	…	0.034031	-0.086308	0.118280	0.083002	
hc_samples	…	0.084131	0.018916	-0.036687	-0.066810	
home_equity_second_mortgage	…	-0.064082	0.015897	-0.013165	-0.009747	
home_equity	…	0.262494	-0.049272	0.195789	0.027243	
debt	…	-0.073428	0.109724	0.013941	-0.047454	
second_mortgage_cdf	…	0.067758	-0.013871	0.086915	0.004593	
hs_degree	…	0.024516	-0.129160	-0.136987	-0.036691	
male_age_mean	…	-0.096096	0.117632	0.093692	0.005857	
male_age_stdev	…	-0.035590	-0.047652	-0.004476	-0.065681	
female_age_stdev	…	0.072675	-0.022256	-0.018799	0.011790	
pct_own	…	0.043028	0.032492	-0.051906	0.085180	
married	…	-0.190224	0.048580	0.075200	0.072399	
married_snp	…	-0.009198	0.033496	0.044966	0.060078	
separated	…	0.067577	-0.076520	-0.109688	-0.066317	
divorced	…	-0.057273	-0.000737	0.022737	0.094977	

	Factor 19	Factor 20	Factor 21	Factor 22	\
ALand	-0.003150	-0.001203	-0.002702	-0.012833	
AWater	0.004022	-0.001757	0.002083	0.019992	
pop	0.051060	0.020066	0.072220	-0.083715	
rent_mean	0.025038	0.000483	-0.041630	0.045671	
rent_stdev	-0.003522	0.005798	0.044094	0.008021	
rent_sample_weight	0.017035	0.014066	-0.030095	0.098533	
rent_gt_10	-0.012662	-0.002203	0.002064	0.015030	
rent_gt_15	-0.000368	0.019973	0.013376	-0.026027	
rent_gt_20	0.009751	-0.002569	-0.009069	0.012176	

hi_mean	0.037871	0.022164	0.009584	0.047005
hi_sample_weight	0.001670	-0.039838	-0.047129	0.027163
hc_mortgage_mean	-0.080945	-0.004473	-0.022017	-0.012636
hc_mortgage_median	-0.133836	-0.023565	-0.043006	-0.012154
hc_mortgage_stdev	0.080331	0.067446	0.030172	0.036778
hc_mortgage_sample_weight	-0.134176	0.146972	-0.037254	0.023278
hc_mean	0.113336	0.084124	0.053171	0.014025
hc_stdev	-0.085390	-0.087805	-0.034870	-0.009250
hc_samples	0.055312	-0.129387	0.020112	-0.008356
home_equity_second_mortgage	-0.016945	0.024014	0.033604	-0.005483
home_equity	-0.011263	0.003748	0.024732	0.010319
debt	0.071367	-0.131799	0.000724	-0.007743
second_mortgage_cdf	-0.022029	0.021478	0.053016	-0.005944
hs_degree	-0.012890	-0.002585	-0.029276	-0.109962
male_age_mean	0.017607	0.022551	-0.039250	-0.024071
male_age_stdev	-0.058146	-0.034938	0.103997	0.023490
female_age_stdev	0.062400	0.025864	-0.119282	-0.052439
pct_own	0.041474	-0.024669	-0.069371	0.114533
married	-0.013776	0.022471	0.081869	-0.016567
married_snp	0.071565	0.044742	-0.065053	-0.047375
separated	-0.067603	-0.047269	0.051104	0.035645
divorced	0.006710	0.011592	0.096582	0.016380

	Factor 23	Factor 24
ALand	-0.014249	0.037596
AWater	0.019599	-0.038246
pop	-0.091054	0.000271
rent_mean	-0.007682	-0.001925
rent_stdev	-0.002431	-0.002180
rent_sample_weight	0.027594	-0.003942
rent_gt_10	-0.015985	0.005271
rent_gt_15	0.039276	-0.002391
rent_gt_20	-0.011723	0.001045
hi_mean	0.079286	0.013039
hi_sample_weight	-0.051184	0.015271
hc_mortgage_mean	-0.008943	0.001656
hc_mortgage_median	0.002378	-0.001406
hc_mortgage_stdev	-0.002918	0.002562
hc_mortgage_sample_weight	0.062471	-0.004938
hc_mean	-0.010532	-0.003617
hc_stdev	0.013653	0.000099
hc_samples	0.108659	-0.007029
home_equity_second_mortgage	0.032678	0.062719
home_equity	0.000213	-0.011839
debt	0.035765	0.014527
second_mortgage_cdf	0.033358	0.071365
hs_degree	-0.026930	0.001279

```
male_age_mean                -0.032915   0.009637
male_age_stdev               -0.017998  -0.007961
female_age_stdev              0.020053   0.016122
pct_own                      -0.084300   0.018359
married                       0.018683  -0.012812
married_snp                   0.043145  -0.010037
separated                    -0.044993   0.008256
divorced                      0.015833  -0.004744

[31 rows x 25 columns]
```

[190]:

```
fa = FactorAnalyzer( rotation="varimax", n_factors = 12)
fa.fit(train_df)
loadings = fa.loadings_
```

[191]:

```
Factors  = pd.DataFrame.from_records(loadings)

Factors = Factors.add_prefix('Factor ')

Factors.index = train_df.columns
Factors
```

[191]:

```
                             Factor 0  Factor 1  Factor 2  Factor 3  Factor 4  \
ALand                       -0.038491 -0.008827  0.020512 -0.074393 -0.026537
AWater                      -0.001059 -0.008456  0.006508 -0.017071 -0.010132
pop                          0.173865  0.885100 -0.072507  0.042089  0.079607
rent_mean                    0.774519  0.001359  0.245323  0.124317  0.098564
rent_stdev                   0.695893 -0.002020  0.028260  0.026617  0.042411
rent_sample_weight          -0.239610  0.239155 -0.819143  0.043269 -0.034753
rent_gt_10                  -0.003201  0.036268 -0.016818  0.611163  0.048986
rent_gt_15                   0.017712  0.001654 -0.037219  0.979854  0.034548
rent_gt_20                  -0.006114 -0.044316 -0.106077  0.741816  0.016374
hi_mean                      0.772987  0.057951  0.387612 -0.153133  0.108461
hi_sample_weight            -0.222135  0.820373 -0.505538  0.088753 -0.004485
hc_mortgage_mean             0.958080 -0.086863  0.026182  0.020973  0.095723
hc_mortgage_median           0.936088 -0.090771  0.029602  0.022791  0.089481
hc_mortgage_stdev            0.782594 -0.037834  0.055870 -0.032568  0.086256
hc_mortgage_sample_weight   -0.281114  0.815470  0.222653 -0.046920  0.080790
hc_mean                      0.842647 -0.073246  0.029399 -0.008721 -0.002358
hc_stdev                     0.709000 -0.019295  0.022643 -0.043465 -0.040643
hc_samples                  -0.083166  0.690176  0.236147 -0.126447 -0.131719
home_equity_second_mortgage  0.032920 -0.016252 -0.046287  0.054894  0.949162
home_equity                  0.422381 -0.027104  0.052724  0.049883  0.524667
debt                         0.324898  0.088280  0.037559  0.138596  0.343328
second_mortgage_cdf         -0.087793 -0.084080 -0.090807 -0.033646 -0.788638
hs_degree                    0.347488  0.084341  0.191827 -0.127091  0.138931
male_age_mean                0.145871 -0.019344  0.221748 -0.098246 -0.064473
```

male_age_stdev	-0.039015	0.048302	0.152456	-0.059910	-0.032291
female_age_stdev	-0.098069	0.035088	0.046868	-0.056899	-0.031312
pct_own	0.077262	0.223320	0.755895	-0.148064	0.053401
married	0.236272	0.227584	0.521015	-0.168636	0.042989
married_snp	-0.051148	-0.045405	-0.111879	0.063161	-0.039003
separated	-0.159963	-0.055347	-0.113873	0.067703	-0.037343
divorced	-0.402441	-0.037581	-0.192822	-0.002475	-0.011628

	Factor 5	Factor 6	Factor 7	Factor 8	Factor 9 \
ALand	-0.001741	0.022035	-0.003820	-0.028440	0.822995
AWater	0.003157	-0.012769	0.007959	-0.007210	0.435603
pop	-0.011951	-0.070458	-0.219569	0.051768	-0.015570
rent_mean	-0.065437	-0.123349	-0.036967	0.160858	-0.020845
rent_stdev	-0.044074	0.016255	-0.002407	0.056207	0.001351
rent_sample_weight	0.114693	-0.045542	-0.031170	0.036964	-0.034004
rent_gt_10	0.011601	-0.060614	-0.015744	0.083916	-0.061191
rent_gt_15	0.039338	-0.037400	-0.014455	0.020597	-0.016987
rent_gt_20	0.105388	-0.022389	-0.068703	-0.051048	-0.025727
hi_mean	-0.218913	0.002787	0.063841	0.236583	-0.011666
hi_sample_weight	0.034825	0.026397	0.085768	-0.054333	-0.021153
hc_mortgage_mean	-0.044628	-0.063441	-0.010632	0.050402	-0.020009
hc_mortgage_median	-0.040252	-0.062975	-0.033270	0.060132	-0.021475
hc_mortgage_stdev	-0.061377	0.020535	0.127077	-0.017269	0.008839
hc_mortgage_sample_weight	-0.150022	0.102130	0.058171	0.224095	-0.023158
hc_mean	-0.076673	-0.037675	0.015651	0.051935	-0.030911
hc_stdev	-0.023018	0.026382	0.080568	-0.091131	0.010325
hc_samples	-0.122456	0.235223	0.301408	-0.410221	0.022862
home_equity_second_mortgage	0.023260	-0.085293	-0.060834	-0.021174	-0.014982
home_equity	-0.130183	-0.014630	0.044343	0.291553	-0.049447
debt	-0.090464	-0.197192	-0.194383	0.704970	-0.082250
second_mortgage_cdf	0.068764	-0.025750	-0.010595	-0.113839	0.015566
hs_degree	-0.393370	0.010532	0.360743	0.295529	0.006618
male_age_mean	-0.069476	0.297650	0.734963	-0.162553	0.011475
male_age_stdev	-0.081228	0.866822	0.126072	-0.058078	0.005281
female_age_stdev	0.006946	0.794347	0.080833	-0.053317	-0.008233
pct_own	-0.269483	0.271089	0.226755	0.038638	0.022699
married	-0.128872	0.324913	0.198060	0.102234	-0.007817
married_snp	0.982863	-0.069050	-0.055987	-0.021407	0.013427
separated	0.667382	-0.011057	0.003131	-0.012628	-0.004970
divorced	0.035919	0.102435	0.308447	-0.009362	0.009847

	Factor 10	Factor 11
ALand	0.016681	0.011140
AWater	-0.010875	-0.005421
pop	0.048907	-0.047273
rent_mean	-0.027264	-0.287506
rent_stdev	-0.092453	-0.236768

```
rent_sample_weight              0.006241   0.067024
rent_gt_10                      0.032058   0.012485
rent_gt_15                      0.006171   0.008518
rent_gt_20                     -0.075981  -0.039883
hi_mean                         0.118342  -0.032497
hi_sample_weight                0.022169  -0.006546
hc_mortgage_mean                0.110707  -0.034964
hc_mortgage_median              0.089246  -0.049569
hc_mortgage_stdev               0.099983   0.033042
hc_mortgage_sample_weight      -0.028230   0.048261
hc_mean                        -0.010578   0.284994
hc_stdev                       -0.067750   0.306172
hc_samples                      0.049124   0.021504
home_equity_second_mortgage     0.002312  -0.014908
home_equity                     0.080034  -0.024319
debt                            0.037043  -0.029291
second_mortgage_cdf             0.017393  -0.010547
hs_degree                       0.002806   0.053311
male_age_mean                  -0.006674   0.000691
male_age_stdev                  0.063422   0.009184
female_age_stdev               -0.049971  -0.002271
pct_own                         0.095988   0.082565
married                         0.461225   0.022392
married_snp                     0.006011  -0.019734
separated                      -0.036025   0.022469
divorced                       -0.376403   0.016110
```

[192]:
```
len(train_df.columns)
```

[192]:
```
31
```

[193]:
```
# Get variance of each factors
fact_variance  = fa.get_factor_variance()
fact_variance
```

[193]:
```
(array([6.22367058, 2.81980847, 2.29931593, 2.08047262, 2.03489583,
        1.83129939, 1.81828516, 1.1108032 , 1.06691425, 0.88807047,
        0.45119573, 0.34422995]),
 array([0.20076357, 0.09096156, 0.07417148, 0.06711202, 0.0656418 ,
        0.05907417, 0.05865436, 0.03583236, 0.03441659, 0.02864743,
        0.0145547 , 0.01110419]),
 array([0.20076357, 0.29172513, 0.36589661, 0.43300863, 0.49865043,
        0.55772461, 0.61637897, 0.65221133, 0.68662792, 0.71527535,
        0.72983005, 0.74093424]))
```

[194]:
```
Factor_variance  = pd.DataFrame.from_records(fact_variance)
```

```python
Factor_variance = Factor_variance.add_prefix('Factor ')

Factor_variance.index = ['SS Loadings', 'Proportion Var', 'Cumulative Var']
round(Factor_variance, 2)
```

[194]:

```
                Factor 0  Factor 1  Factor 2  Factor 3  Factor 4  Factor 5  \
SS Loadings         6.22      2.82      2.30      2.08      2.03      1.83
Proportion Var      0.20      0.09      0.07      0.07      0.07      0.06
Cumulative Var      0.20      0.29      0.37      0.43      0.50      0.56

                Factor 6  Factor 7  Factor 8  Factor 9  Factor 10  Factor 11
SS Loadings         1.82      1.11      1.07      0.89       0.45       0.34
Proportion Var      0.06      0.04      0.03      0.03       0.01       0.01
Cumulative Var      0.62      0.65      0.69      0.72       0.73       0.74
```

[ ]: