Capstone Project_1- Real Estate.

In [59]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.decomposition import FactorAnalysis
from factor_analyzer import FactorAnalyzer
```

In [3]:
```
df_train=pd.read_csv('train.csv')
df_test=pd.read_csv('test.csv')
```

In [4]:
```
df_train.head()
```

Out[4]:

	UID	BLOCKID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	type	...	female_age_mean	female_age_median	female_age_s
0	267822	NaN	140	53	36	New York	NY	Hamilton	Hamilton	City	...	44.48629	45.33333	22.5
1	246444	NaN	140	141	18	Indiana	IN	South Bend	Roseland	City	...	36.48391	37.58333	23.4
2	245683	NaN	140	63	18	Indiana	IN	Danville	Danville	City	...	42.15810	42.83333	23.9
3	279653	NaN	140	127	72	Puerto Rico	PR	San Juan	Guaynabo	Urban	...	47.77526	50.58333	24.3
4	247218	NaN	140	161	20	Kansas	KS	Manhattan	Manhattan City	City	...	24.17693	21.58333	11.1

5 rows × 80 columns

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

Out[5]:

	UID	BLOCKID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	type	...	female_age_mean	female_age_median	female
0	255504	NaN	140	163	26	Michigan	MI	Detroit	Dearborn Heights City	CDP	...	34.78682	33.75000	
1	252676	NaN	140	1	23	Maine	ME	Auburn	Auburn City	City	...	44.23451	46.66667	
2	276314	NaN	140	15	42	Pennsylvania	PA	Pine City	Millerton	Borough	...	41.62426	44.50000	
3	248614	NaN	140	231	21	Kentucky	KY	Monticello	Monticello City	City	...	44.81200	48.00000	
4	286865	NaN	140	355	48	Texas	TX	Corpus Christi	Edroy	Town	...	40.66618	42.66667	

5 rows × 80 columns

In [6]:
```
df_train.shape
```

Out[6]:
```
(27321, 80)
```

In [7]:
```
df_test.shape
```

Out[7]:
```
(11709, 80)
```

In [8]:
```
len(set(df_train['UID']).intersection(set(df_test['UID'])))
```

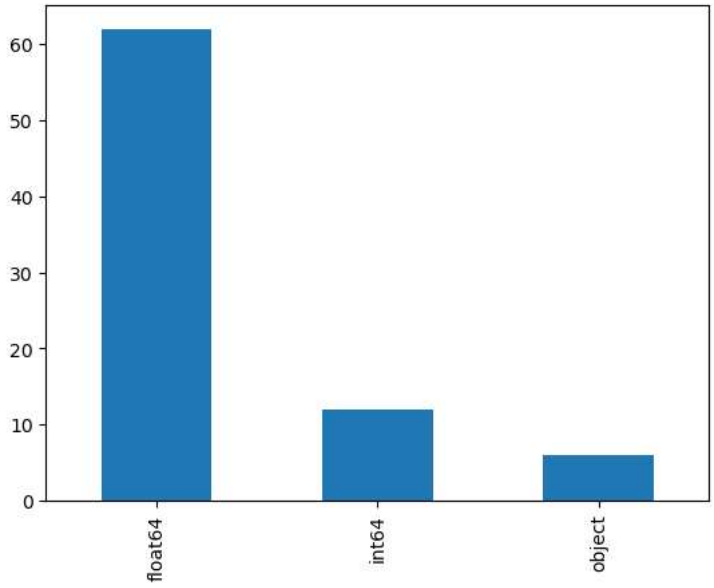
Out[8]:
```
123
```

In [9]:
```
df_train.dtypes
```

Out[9]:
```
UID             int64
BLOCKID       float64
SUMLEVEL        int64
COUNTYID        int64
STATEID         int64
               ...
pct_own       float64
married       float64
married_snp   float64
separated     float64
divorced      float64
Length: 80, dtype: object
```

In [10]:
```
df_train.dtypes.value_counts().plot(kind='bar')
```

Out[10]: <AxesSubplot:>



In [11]:
```
df_train.describe(include='O')
```

Out[11]:

	state	state_ab	city	place	type	primary
count	27321	27321	27321	27321	27321	27321
unique	52	52	6916	9912	6	1
top	California	CA	Chicago	New York City	City	tract
freq	2926	2926	294	490	15237	27321

In [12]:
```
df_train['split']= 'Train'
df_test['split']= 'Test'
```

In [13]:
```
df_combined=df_train.append(df_test, ignore_index=True)
df_combined.head()
```

```
C:\Users\S Singh\AppData\Local\Temp\ipykernel_2440\1695772958.py:1: FutureWarning: The frame.append method is deprecated and wi
ll be removed from pandas in a future version. Use pandas.concat instead.
  df_combined=df_train.append(df_test, ignore_index=True)
```

Out[13]:

	UID	BLOCKID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	type	...	female_age_median	female_age_stdev	female_age_s
0	267822	NaN	140	53	36	New York	NY	Hamilton	Hamilton	City	...	45.33333	22.51276	
1	246444	NaN	140	141	18	Indiana	IN	South Bend	Roseland	City	...	37.58333	23.43353	
2	245683	NaN	140	63	18	Indiana	IN	Danville	Danville	City	...	42.83333	23.94119	
3	279653	NaN	140	127	72	Puerto Rico	PR	San Juan	Guaynabo	Urban	...	50.58333	24.32015	
4	247218	NaN	140	161	20	Kansas	KS	Manhattan	Manhattan City	City	...	21.58333	11.10484	

5 rows × 81 columns

In [14]:
```
df_combined.tail()
```

Out[14]:

	UID	BLOCKID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	type	...	female_age_median	female_age_stdev	fem
39025	238088	NaN	140	105	12	Florida	FL	Lakeland	Crystal Springs	City	...	59.58333	23.23426	
39026	242811	NaN	140	31	17	Illinois	IL	Chicago	Chicago City	Village	...	32.83333	20.24698	
39027	250127	NaN	140	9	25	Massachusetts	MA	Lawrence	Methuen Town City	City	...	43.66667	23.17995	
39028	241096	NaN	140	27	19	Iowa	IA	Carroll	Carroll City	City	...	48.16667	24.84209	
39029	287763	NaN	140	453	48	Texas	TX	Austin	Sunset Valley City	Town	...	35.41667	20.68049	

5 rows × 81 columns

In [15]:
```
df_combined.shape
```

Out[15]:
```
(39030, 81)
```

In [16]:
```
df_combined.isna().sum()
```

Out[16]:
```
UID               0
BLOCKID       39030
SUMLEVEL          0
COUNTYID          0
STATEID           0
              ...
married         275
married_snp     275
separated       275
divorced        275
split             0
Length: 81, dtype: int64
```

In [17]:
```
1-df_combined.isna().sum()/len(df_combined)
```

Out[17]:
```
UID           1.000000
BLOCKID       0.000000
SUMLEVEL      1.000000
COUNTYID      1.000000
STATEID       1.000000
                ...
married       0.992954
married_snp   0.992954
separated     0.992954
divorced      0.992954
split         1.000000
Length: 81, dtype: float64
```

In [18]:
```
# BLOCKID is completly missing in both train and test data. So we will drop BLOCKID feature.
df_combined.drop(columns =['BLOCKID'], axis=1, inplace=True)
```

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

Out[19]:
```
UID           0.000000
SUMLEVEL      0.000000
COUNTYID      0.000000
STATEID       0.000000
state         0.000000
                ...
married       0.704586
married_snp   0.704586
separated     0.704586
divorced      0.704586
split         0.000000
Length: 80, dtype: float64
```

In [20]:
```
# Missing value greater than zero
col_check=df_combined.isna().sum().to_frame().reset_index()
null_col=col_check[col_check[0]>0]['index'].tolist()
null_col
```

Out[20]:
```
['rent_mean',
 'rent_median',
 'rent_stdev',
 'rent_sample_weight',
 'rent_samples',
 'rent_gt_10',
 'rent_gt_15',
 'rent_gt_20',
 'rent_gt_25',
 'rent_gt_30',
 'rent_gt_35',
 'rent_gt_40',
 'rent_gt_50',
 'hi_mean',
 'hi_median',
 'hi_stdev',
 'hi_sample_weight',
 'hi_samples',
 'family_mean',
 'family_median',
 'family_stdev',
 'family_sample_weight',
 'family_samples',
 'hc_mortgage_mean',
 'hc_mortgage_median',
 'hc_mortgage_stdev',
 'hc_mortgage_sample_weight',
 'hc_mortgage_samples',
 'hc_mean',
 'hc_median',
 'hc_stdev',
 'hc_samples',
 'hc_sample_weight',
 'home_equity_second_mortgage',
 'second_mortgage',
 'home_equity',
 'debt',
 'second_mortgage_cdf',
 'home_equity_cdf',
 'debt_cdf',
 'hs_degree',
 'hs_degree_male',
 'hs_degree_female',
 'male_age_mean',
 'male_age_median',
 'male_age_stdev',
 'male_age_sample_weight',
 'male_age_samples',
 'female_age_mean',
 'female_age_median',
 'female_age_stdev',
 'female_age_sample_weight',
 'female_age_samples',
 'pct_own',
 'married',
 'married_snp',
 'separated',
 'divorced']
```

In [21]:
```
#If the feature have less than 8 unique value then I am consdering as categorical else it will be continuous
for i in null_col:
    print(i)
    if df_combined[i].nunique()>8:      #Continuous data
        df_combined[i].fillna(df_combined[i].median(),inplace=True)    #Bcz median is not impacted by outlier
    else:df_combined[i].fillna(df_combined[i].mode()[0],inplace=True)
```

```
rent_mean
rent_median
rent_stdev
rent_sample_weight
rent_samples
rent_gt_10
rent_gt_15
rent_gt_20
rent_gt_25
rent_gt_30
rent_gt_35
rent_gt_40
rent_gt_50
hi_mean
hi_median
hi_stdev
hi_sample_weight
hi_samples
family_mean
family_median
family_stdev
family_sample_weight
family_samples
hc_mortgage_mean
hc_mortgage_median
hc_mortgage_stdev
hc_mortgage_sample_weight
hc_mortgage_samples
hc_mean
hc_median
hc_stdev
hc_samples
hc_sample_weight
home_equity_second_mortgage
second_mortgage
home_equity
debt
second_mortgage_cdf
home_equity_cdf
debt_cdf
hs_degree
hs_degree_male
hs_degree_female
male_age_mean
male_age_median
male_age_stdev
male_age_sample_weight
male_age_samples
female_age_mean
female_age_median
female_age_stdev
female_age_sample_weight
female_age_samples
pct_own
married
married_snp
separated
divorced
```

In [22]:
```
df_combined.isna().sum()/len(df_combined)*100
```

Out[22]:
```
UID             0.0
SUMLEVEL        0.0
COUNTYID        0.0
STATEID         0.0
state           0.0
               ...
married         0.0
married_snp     0.0
separated       0.0
divorced        0.0
split           0.0
Length: 80, dtype: float64
```

In [23]:
```
df_combined.shape
```

Out[23]: (39030, 80)

In [24]:
```
# Drop duplicate observations
df_combined.drop_duplicates(inplace=True)
df_combined.shape
```

Out[24]: (38838, 80)

In [25]:
```
# As we have seen above we have 123 unique UID which are common in both train and test data. so duplicate UID removing them.
df_combined.drop_duplicates(subset=['UID'],inplace=True)
df_combined.shape
```

Out[25]: (38715, 80)

Exploratory Data Analysis (EDA):

In [26]:
```
top_2500_loc=df_train[(df_train['second_mortgage']<0.50) &
                      (df_train['pct_own']>0.10) ].sort_values(by='second_mortgage', ascending=False).head(2500)
```

In [27]:
```
top_2500_loc=top_2500_loc[['state','city','state_ab','place','lat','lng']]
top_2500_loc.head()
```

Out[27]:

	state	city	state_ab	place	lat	lng
11980	Massachusetts	Worcester	MA	Worcester City	42.254262	-71.800347
26018	New York	Corona	NY	Harbor Hills	40.751809	-73.853582
7829	Maryland	Glen Burnie	MD	Glen Burnie	39.127273	-76.635265
2077	Florida	Tampa	FL	Egypt Lake-leto	28.029063	-82.495395
1701	Illinois	Chicago	IL	Lincolnwood	41.967289	-87.652434

In [28]:
```
!pip install geopandas
import warnings
warnings.filterwarnings('ignore')
```

```
Collecting geopandas
  Downloading geopandas-0.14.0-py3-none-any.whl (1.1 MB)
     ---------------------------------------- 1.1/1.1 MB 1.6 MB/s eta 0:00:00
Requirement already satisfied: pandas>=1.4.0 in c:\users\s singh\anaconda3\lib\site-packages (from geopandas) (1.4.4)
Collecting fiona>=1.8.21
  Downloading fiona-1.9.5-cp39-cp39-win_amd64.whl (22.9 MB)
     ---------------------------------------- 22.9/22.9 MB 2.7 MB/s eta 0:00:00
Collecting pyproj>=3.3.0
  Downloading pyproj-3.6.1-cp39-cp39-win_amd64.whl (6.1 MB)
     ---------------------------------------- 6.1/6.1 MB 2.7 MB/s eta 0:00:00
Collecting shapely>=1.8.0
  Downloading shapely-2.0.2-cp39-cp39-win_amd64.whl (1.4 MB)
     ---------------------------------------- 1.4/1.4 MB 2.3 MB/s eta 0:00:00
Requirement already satisfied: packaging in c:\users\s singh\anaconda3\lib\site-packages (from geopandas) (21.3)
Requirement already satisfied: setuptools in c:\users\s singh\anaconda3\lib\site-packages (from fiona>=1.8.21->geopandas) (63.
4.1)
Collecting cligj>=0.5
  Downloading cligj-0.7.2-py3-none-any.whl (7.1 kB)
Requirement already satisfied: certifi in c:\users\s singh\anaconda3\lib\site-packages (from fiona>=1.8.21->geopandas) (2022.9.
14)
Requirement already satisfied: attrs>=19.2.0 in c:\users\s singh\anaconda3\lib\site-packages (from fiona>=1.8.21->geopandas) (2
1.4.0)
Requirement already satisfied: importlib-metadata in c:\users\s singh\anaconda3\lib\site-packages (from fiona>=1.8.21->geopanda
s) (4.11.3)
Requirement already satisfied: six in c:\users\s singh\anaconda3\lib\site-packages (from fiona>=1.8.21->geopandas) (1.16.0)
Requirement already satisfied: click~=8.0 in c:\users\s singh\anaconda3\lib\site-packages (from fiona>=1.8.21->geopandas) (8.0.
4)
Collecting click-plugins>=1.0
  Downloading click_plugins-1.1.1-py2.py3-none-any.whl (7.5 kB)
Requirement already satisfied: numpy>=1.18.5 in c:\users\s singh\anaconda3\lib\site-packages (from pandas>=1.4.0->geopandas)
(1.21.5)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\s singh\anaconda3\lib\site-packages (from pandas>=1.4.0->geop
andas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\s singh\anaconda3\lib\site-packages (from pandas>=1.4.0->geopandas) (20
22.1)
Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in c:\users\s singh\anaconda3\lib\site-packages (from packaging->geopan
das) (3.0.9)
Requirement already satisfied: colorama in c:\users\s singh\anaconda3\lib\site-packages (from click~=8.0->fiona>=1.8.21->geopan
das) (0.4.5)
Requirement already satisfied: zipp>=0.5 in c:\users\s singh\anaconda3\lib\site-packages (from importlib-metadata->fiona>=1.8.2
1->geopandas) (3.8.0)
Installing collected packages: shapely, pyproj, cligj, click-plugins, fiona, geopandas
Successfully installed click-plugins-1.1.1 cligj-0.7.2 fiona-1.9.5 geopandas-0.14.0 pyproj-3.6.1 shapely-2.0.2
```

In [29]:
```
import geopandas as gpd
gdf = gpd.GeoDataFrame(top_2500_loc, geometry=gpd.points_from_xy(x=top_2500_loc.lng, y=top_2500_loc.lat))
gdf
```

Out[29]:

	state	city	state_ab	place	lat	lng	geometry
11980	Massachusetts	Worcester	MA	Worcester City	42.254262	-71.800347	POINT (-71.80035 42.25426)
26018	New York	Corona	NY	Harbor Hills	40.751809	-73.853582	POINT (-73.85358 40.75181)
7829	Maryland	Glen Burnie	MD	Glen Burnie	39.127273	-76.635265	POINT (-76.63526 39.12727)
2077	Florida	Tampa	FL	Egypt Lake-leto	28.029063	-82.495395	POINT (-82.49540 28.02906)
1701	Illinois	Chicago	IL	Lincolnwood	41.967289	-87.652434	POINT (-87.65243 41.96729)
...	...	...	...	...	...	...	...
17914	North Carolina	Raleigh	NC	Raleigh City	35.757135	-78.704288	POINT (-78.70429 35.75713)
5478	California	Marina Del Rey	CA	Marina Del Rey	33.983204	-118.466139	POINT (-118.46614 33.98320)
25642	Maryland	Baltimore	MD	Lochearn	39.353095	-76.733315	POINT (-76.73331 39.35310)
26671	Pennsylvania	Philadelphia	PA	Philadelphia City	40.039070	-75.125135	POINT (-75.12514 40.03907)
24443	California	Manteca	CA	Manteca City	37.732143	-121.242902	POINT (-121.24290 37.73214)

2500 rows × 7 columns

In [30]:
```
#Bad Debt = second_mortgage + home_equity - home_equity_second_mortgage
df_combined['bad_debt'] = df_combined['second_mortgage'] + df_combined['home_equity'] - df_combined['home_equity_second_mortgage']
df_combined.head()
```

Out[30]:

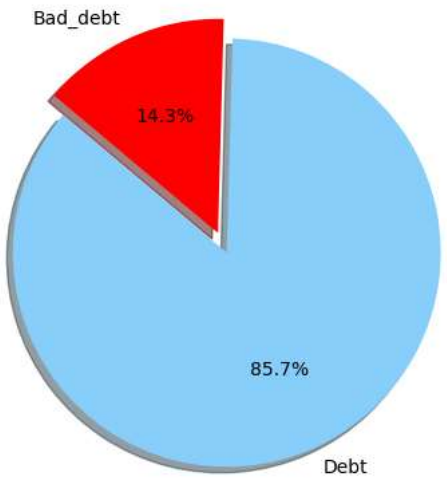
	UID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	type	primary	...	female_age_stdev	female_age_sample_weight	female_
0	267822	140	53	36	New York	NY	Hamilton	Hamilton	City	tract	...	22.51276	685.33845	
1	246444	140	141	18	Indiana	IN	South Bend	Roseland	City	tract	...	23.43353	267.23367	
2	245683	140	63	18	Indiana	IN	Danville	Danville	City	tract	...	23.94119	707.01963	
3	279653	140	127	72	Puerto Rico	PR	San Juan	Guaynabo	Urban	tract	...	24.32015	362.20193	
4	247218	140	161	20	Kansas	KS	Manhattan	Manhattan City	City	tract	...	11.10484	1854.48652	

5 rows × 81 columns

In [31]:
```
labels = 'Debt', 'Bad_debt'
sizes = [df_combined['debt'].mean()*100, df_combined['bad_debt'].mean()*100]
colors = [ 'lightskyblue','red']
explode = (0.1, 0)  # explode 1st slice

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

plt.axis('equal')
plt.show()
```



In [32]:
```
df_combined['good_debt']=df_combined['debt']-df_combined['bad_debt']
df_combined.head()
```

Out[32]:

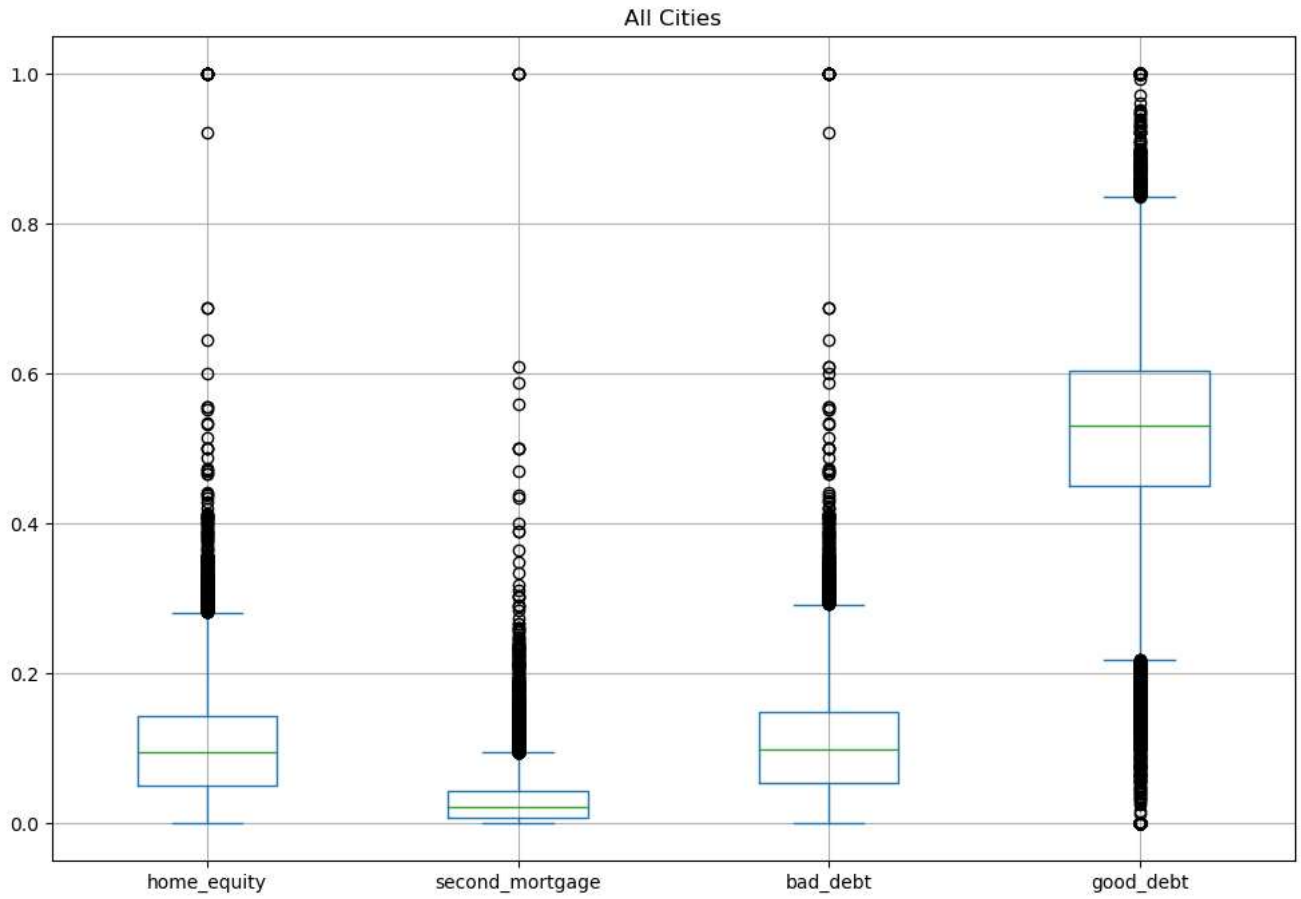
	UID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	type	primary	...	female_age_sample_weight	female_age_samples	pct_
0	267822	140	53	36	New York	NY	Hamilton	Hamilton	City	tract	...	685.33845	2618.0	0.79
1	246444	140	141	18	Indiana	IN	South Bend	Roseland	City	tract	...	267.23367	1284.0	0.52
2	245683	140	63	18	Indiana	IN	Danville	Danville	City	tract	...	707.01963	3238.0	0.85
3	279653	140	127	72	Puerto Rico	PR	San Juan	Guaynabo	Urban	tract	...	362.20193	1559.0	0.65
4	247218	140	161	20	Kansas	KS	Manhattan	Manhattan City	City	tract	...	1854.48652	3051.0	0.13

5 rows × 82 columns

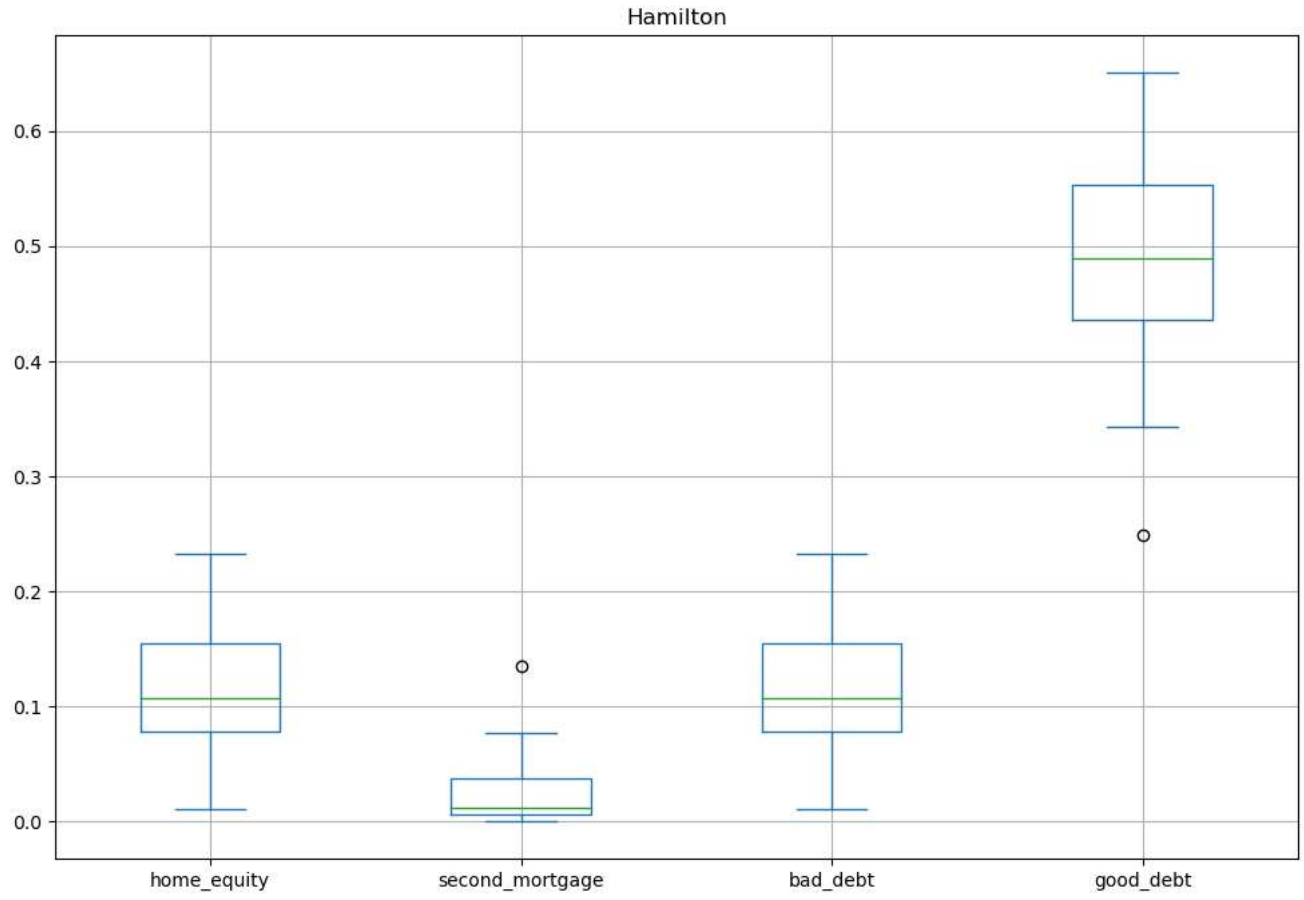
In [33]:
```
df_combined.columns
```

Out[33]:
```
Index(['UID', 'SUMLEVEL', 'COUNTYID', 'STATEID', 'state', 'state_ab', 'city',
       'place', 'type', 'primary', 'zip_code', 'area_code', 'lat', 'lng',
       'ALand', 'AWater', 'pop', 'male_pop', 'female_pop', 'rent_mean',
       'rent_median', 'rent_stdev', 'rent_sample_weight', 'rent_samples',
       'rent_gt_10', 'rent_gt_15', 'rent_gt_20', 'rent_gt_25', 'rent_gt_30',
       'rent_gt_35', 'rent_gt_40', 'rent_gt_50', 'universe_samples',
       'used_samples', 'hi_mean', 'hi_median', 'hi_stdev', 'hi_sample_weight',
       'hi_samples', 'family_mean', 'family_median', 'family_stdev',
       'family_sample_weight', 'family_samples', 'hc_mortgage_mean',
       'hc_mortgage_median', 'hc_mortgage_stdev', 'hc_mortgage_sample_weight',
       'hc_mortgage_samples', 'hc_mean', 'hc_median', 'hc_stdev', 'hc_samples',
       'hc_sample_weight', 'home_equity_second_mortgage', 'second_mortgage',
       'home_equity', 'debt', 'second_mortgage_cdf', 'home_equity_cdf',
       'debt_cdf', 'hs_degree', 'hs_degree_male', 'hs_degree_female',
       'male_age_mean', 'male_age_median', 'male_age_stdev',
       'male_age_sample_weight', 'male_age_samples', 'female_age_mean',
       'female_age_median', 'female_age_stdev', 'female_age_sample_weight',
       'female_age_samples', 'pct_own', 'married', 'married_snp', 'separated',
       'divorced', 'split', 'bad_debt', 'good_debt'],
      dtype='object')
```

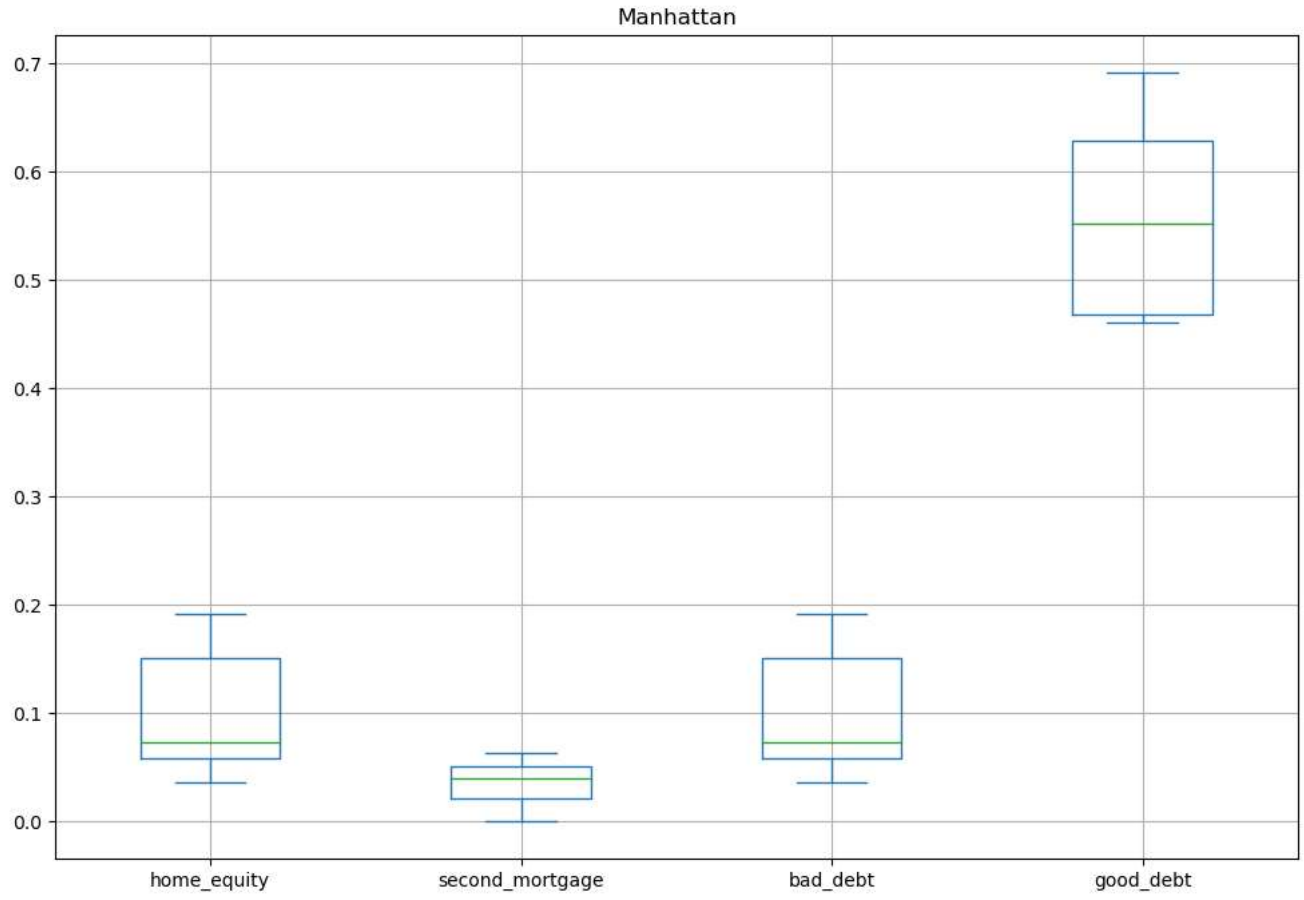
In [34]:
```
all_cities = df_combined[['home_equity','second_mortgage','bad_debt', 'good_debt']]
all_cities.plot.box(figsize=(12,8),grid=True)
plt.title('All Cities')
plt.show()
```



In [35]:
```
hamilton = df_combined[df_combined['city']=='Hamilton']
hamilton = hamilton[['home_equity','second_mortgage','bad_debt', 'good_debt']]
hamilton.plot.box(figsize=(12,8),grid=True)
plt.title('Hamilton')
plt.show()
```

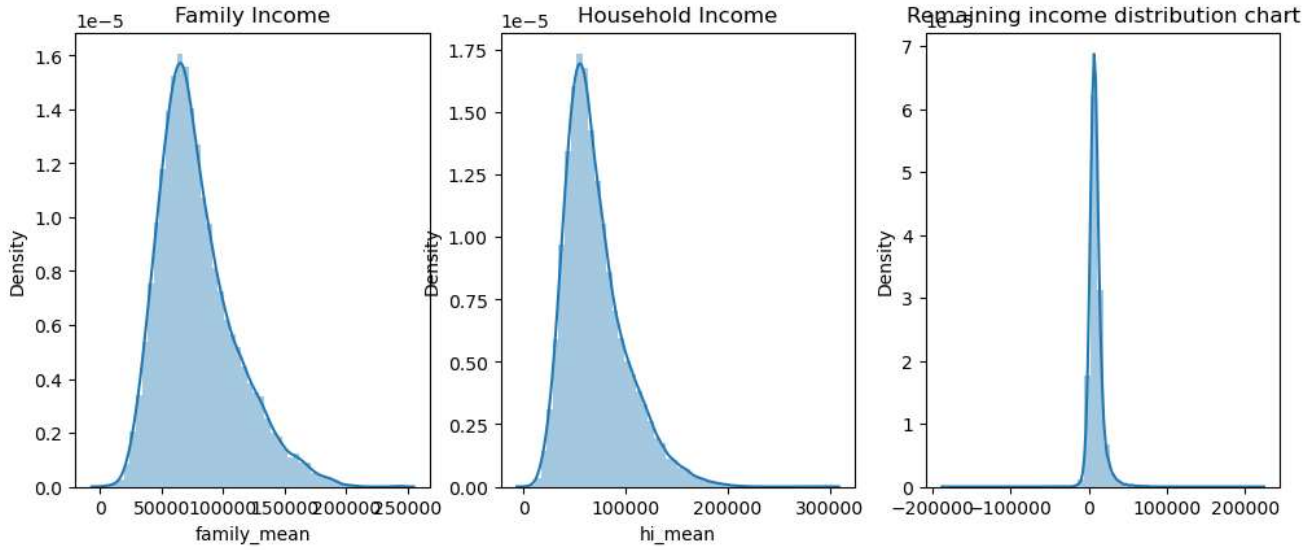


In [36]:
```
Manhattan = df_combined[df_combined['city']=='Manhattan']
Manhattan = Manhattan[['home_equity','second_mortgage','bad_debt', 'good_debt']]
Manhattan.plot.box(figsize=(12,8),grid=True)
plt.title('Manhattan')
plt.show()
```



In [37]:
```
import seaborn as sns
plt.figure(figsize=(12,10))

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



In [38]:
```
df_combined['population_density'] = df_combined['pop']/df_combined['ALand']
```

In [39]:
```
df_combined.head()
```

Out[39]:

	UID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	type	primary	...	female_age_samples	pct_own	married	married_snp
0	267822	140	53	36	New York	NY	Hamilton	Hamilton	City	tract	...	2618.0	0.79046	0.57851	0.01882
1	246444	140	141	18	Indiana	IN	South Bend	Roseland	City	tract	...	1284.0	0.52483	0.34886	0.01426
2	245683	140	63	18	Indiana	IN	Danville	Danville	City	tract	...	3238.0	0.85331	0.64745	0.02830
3	279653	140	127	72	Puerto Rico	PR	San Juan	Guaynabo	Urban	tract	...	1559.0	0.65037	0.47257	0.02021
4	247218	140	161	20	Kansas	KS	Manhattan	Manhattan City	City	tract	...	3051.0	0.13046	0.12356	0.00000

5 rows × 83 columns

In [40]:
```
# Weighted average
df_combined['median_age']=((df_combined['male_age_median'] * df_combined['male_pop'])+(df_combined['female_age_median']*df_combir
```

In [41]:
```
df_combined.head()
```

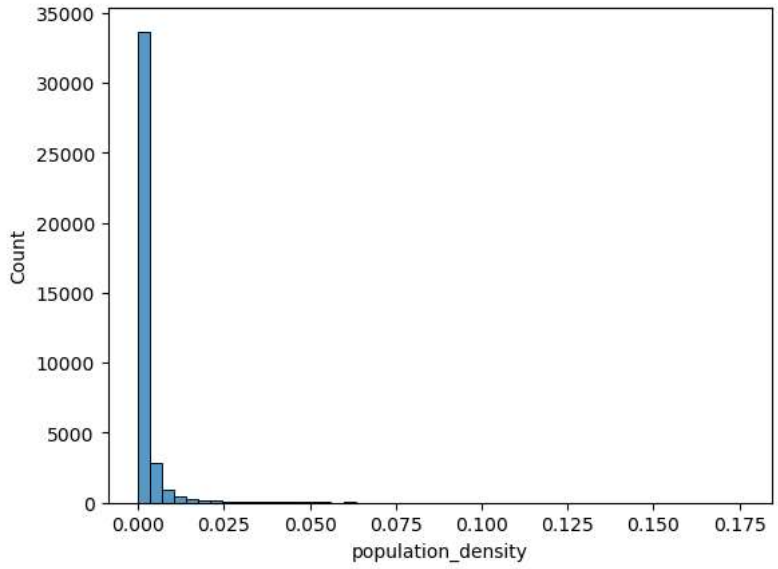
Out[41]:

	UID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	type	primary	...	pct_own	married	married_snp	separated	divorced
0	267822	140	53	36	New York	NY	Hamilton	Hamilton	City	tract	...	0.79046	0.57851	0.01882	0.01240	0.08770
1	246444	140	141	18	Indiana	IN	South Bend	Roseland	City	tract	...	0.52483	0.34886	0.01426	0.01426	0.09030
2	245683	140	63	18	Indiana	IN	Danville	Danville	City	tract	...	0.85331	0.64745	0.02830	0.01607	0.10657
3	279653	140	127	72	Puerto Rico	PR	San Juan	Guaynabo	Urban	tract	...	0.65037	0.47257	0.02021	0.02021	0.10106
4	247218	140	161	20	Kansas	KS	Manhattan	Manhattan City	City	tract	...	0.13046	0.12356	0.00000	0.00000	0.03109

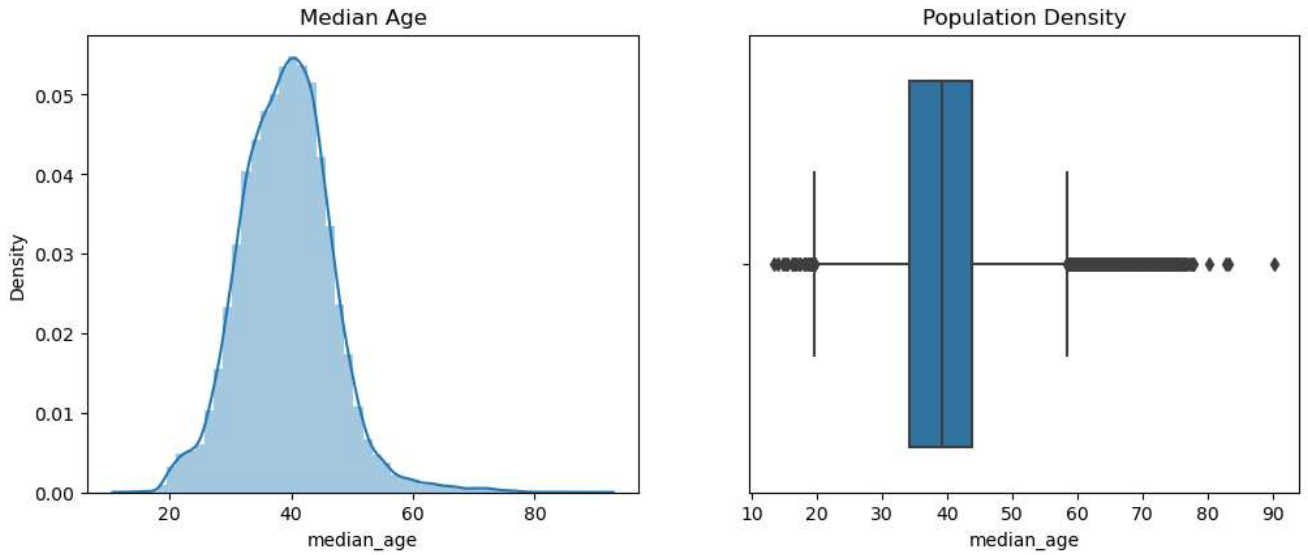
5 rows × 84 columns

In [42]:
```
sns.histplot(df_combined['population_density'], bins=50)
```

Out[42]:
```
<AxesSubplot:xlabel='population_density', ylabel='Count'>
```



In [43]:
```
plt.figure(figsize=(12,10))
plt.subplot(2,2,1)
sns.distplot(df_combined['median_age'])
plt.title('Median Age')
plt.subplot(2,2,2)
sns.boxplot(df_combined['median_age'])
plt.title('Population Density')
plt.show()
```



In [44]:
```
df_combined['pop_bins']=pd.cut(df_combined['pop'],bins=5,labels=['very low','low','medium','high','very high'])
df_combined['pop_bins'].value_counts()
```

Out[44]:
```
very low     38350
low            348
medium          12
high             4
very high        1
Name: pop_bins, dtype: int64
```

In [45]:
```
df_combined.groupby(by='pop_bins')[['married','separated','divorced']].count()
```

Out[45]:

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

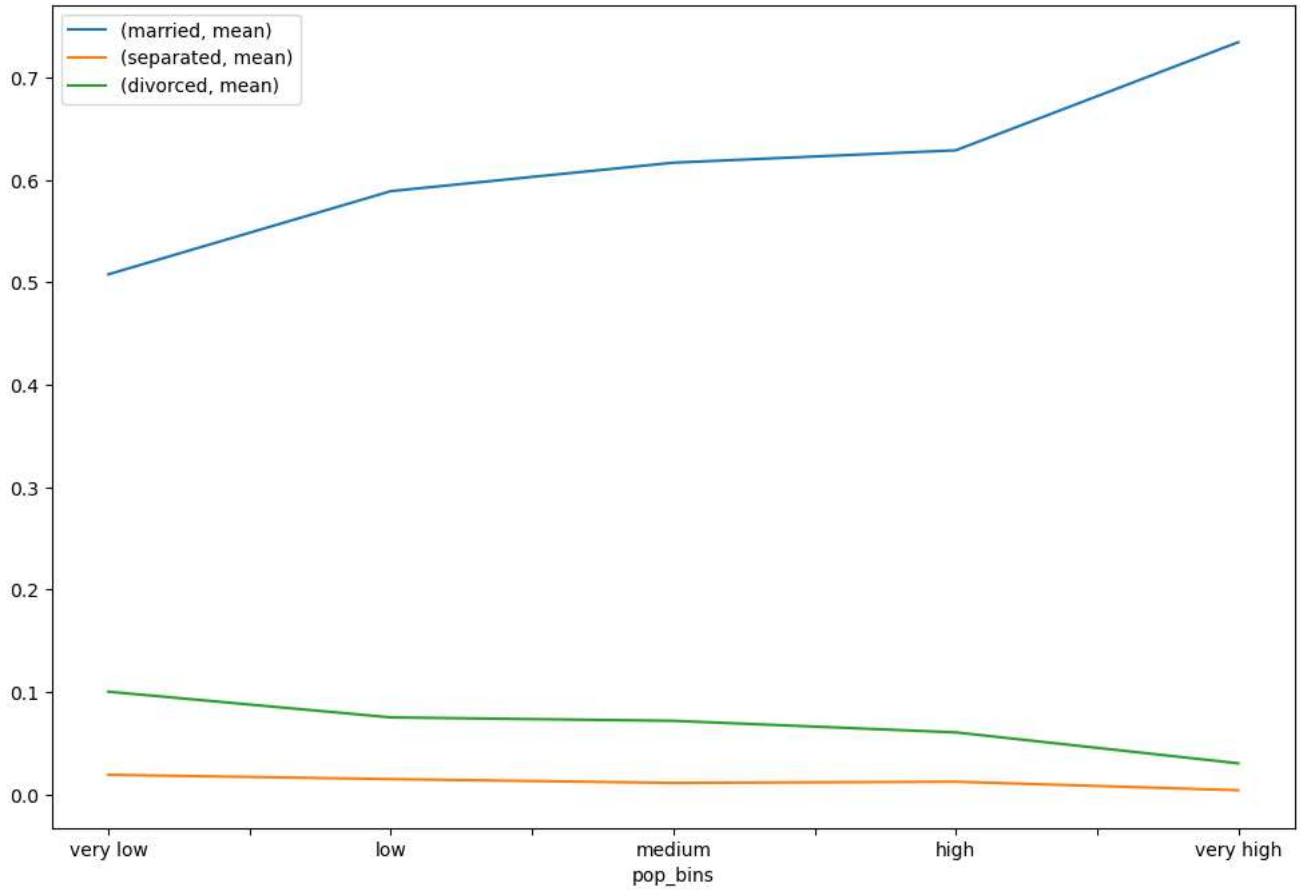
In [46]:
```
df_combined.groupby(by='pop_bins')[['married','separated','divorced']].agg(["mean", "median"])
```

Out[46]:

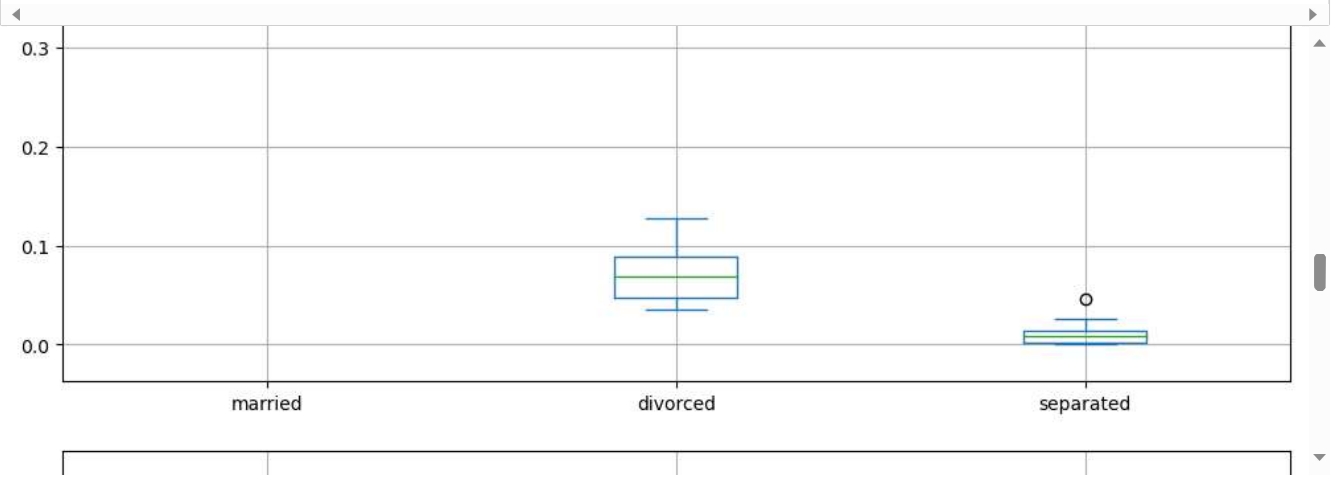
	married		separated		divorced	
pop_bins	mean	median	mean	median	mean	median
very low	0.508000	0.526210	0.019127	0.013580	0.100325	0.09510
low	0.589247	0.601815	0.014929	0.010255	0.075192	0.06934
medium	0.617047	0.605765	0.011203	0.007745	0.071870	0.06909
high	0.629132	0.675095	0.012372	0.007340	0.060562	0.05987
very high	0.734740	0.734740	0.004050	0.004050	0.030360	0.03036

In [47]:
```
plt.figure(figsize=(12,8))
pop_bin_married=df_combined.groupby(by='pop_bins')[['married','separated','divorced']].agg(["mean"])
pop_bin_married.plot(figsize=(12,8))
plt.legend(loc='best')
plt.show()
```

```
<Figure size 1200x800 with 0 Axes>
```



In [48]:
```
df_combined.groupby(by='pop_bins')[['married','divorced', 'separated']].plot.box(figsize=(12,8),grid='True')
plt.show()
```



In [49]:
```
rent_state_mean = df_combined.groupby(by='state')['rent_mean'].agg(["mean"])
rent_state_mean.head()
```

Out[49]:

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

In [50]:
```
income_state_mean=df_combined.groupby(by='state')['family_mean'].agg(["mean"])
income_state_mean.head()
```

Out[50]:

state	mean
Alabama	65311.510962
Alaska	91911.137520
Arizona	73014.068487
Arkansas	64234.705963
California	87711.550734

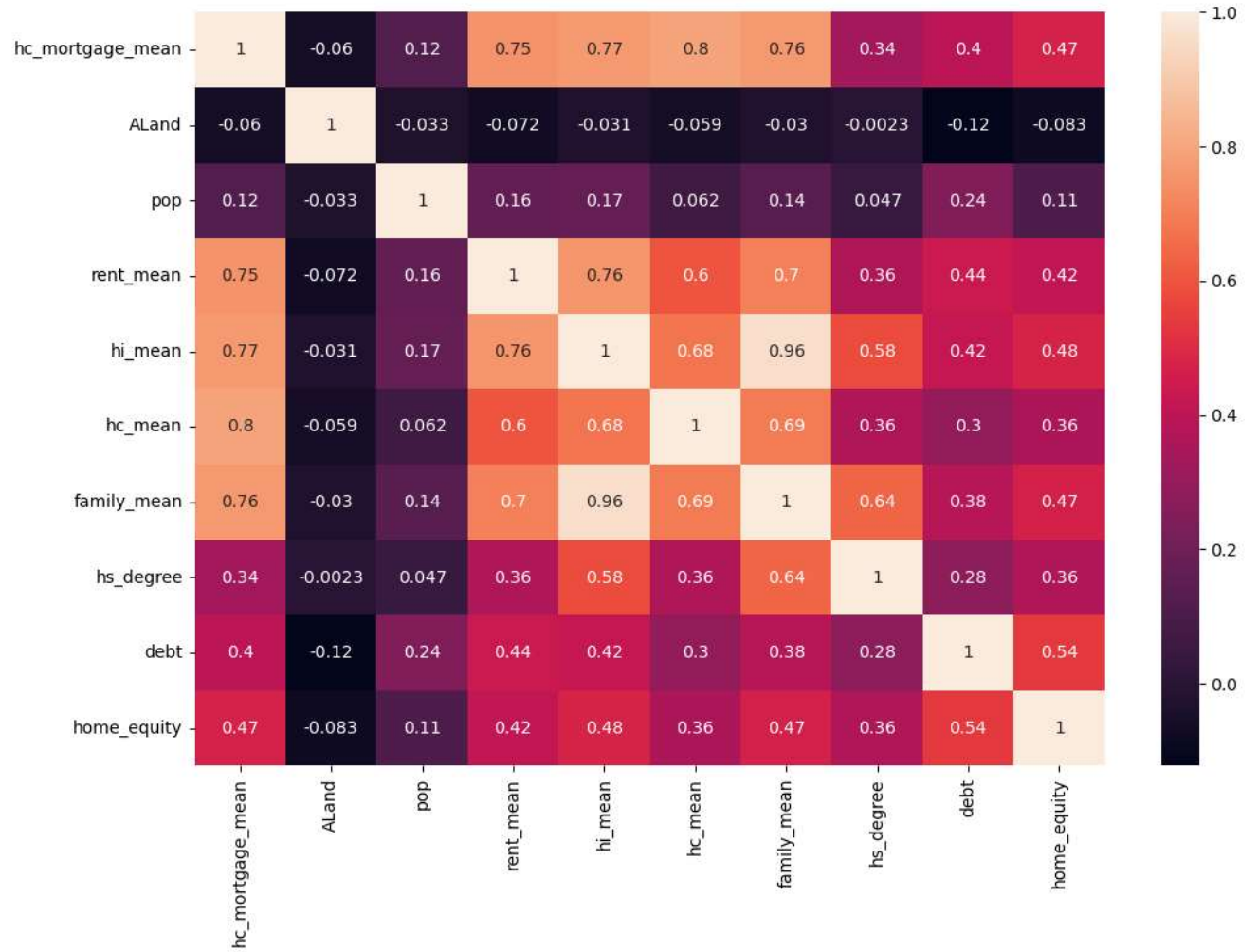
In [51]:
```
rent_perc_of_income=rent_state_mean['mean']/income_state_mean['mean']*100
rent_perc_of_income.head(10)
```

Out[51]:
```
state
Alabama                 1.172646
Alaska                  1.294831
Arizona                 1.485345
Arkansas                1.115511
California              1.671411
Colorado                1.359697
Connecticut             1.272141
Delaware                1.311538
District of Columbia    1.357450
Florida                 1.576101
Name: mean, dtype: float64
```

In [52]:
```
sum(df_combined['rent_mean'])/sum(df_combined['family_mean'])
```

Out[52]:
```
0.013351543786573208
```

In [53]:
```
plt.figure(figsize=(12,8))
sns.heatmap(data=df_combined[['hc_mortgage_mean','ALand','pop','rent_mean','hi_mean','hc_mean','family_mean',
                              'hs_degree','debt','home_equity']].corr(),annot=True)
plt.show()
```



In [54]:
```
train = df_combined[df_combined['split'] == 'Train']
test = df_combined[df_combined['split'] == 'Test']
```

In [55]:
```
train.head()
```

Out[55]:

	UID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	type	primary	...	married	married_snp	separated	divorced	split	bac
0	267822	140	53	36	New York	NY	Hamilton	Hamilton	City	tract	...	0.57851	0.01882	0.01240	0.08770	Train	0
1	246444	140	141	18	Indiana	IN	South Bend	Roseland	City	tract	...	0.34886	0.01426	0.01426	0.09030	Train	0
2	245683	140	63	18	Indiana	IN	Danville	Danville	City	tract	...	0.64745	0.02830	0.01607	0.10657	Train	0
3	279653	140	127	72	Puerto Rico	PR	San Juan	Guaynabo	Urban	tract	...	0.47257	0.02021	0.02021	0.10106	Train	0
4	247218	140	161	20	Kansas	KS	Manhattan	Manhattan City	City	tract	...	0.12356	0.00000	0.00000	0.03109	Train	0

5 rows × 85 columns

In [56]:
```
test.head()
```

Out[56]:

	UID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	type	primary	...	married	married_snp	separated	divorced
27321	255504	140	163	26	Michigan	MI	Detroit	Dearborn Heights City	CDP	tract	...	0.28217	0.05910	0.03813	0.14299
27322	252676	140	1	23	Maine	ME	Auburn	Auburn City	City	tract	...	0.64221	0.02338	0.00000	0.13377
27323	276314	140	15	42	Pennsylvania	PA	Pine City	Millerton	Borough	tract	...	0.59961	0.01746	0.01358	0.10026
27324	248614	140	231	21	Kentucky	KY	Monticello	Monticello City	City	tract	...	0.56953	0.05492	0.04694	0.12489
27325	286865	140	355	48	Texas	TX	Corpus Christi	Edroy	Town	tract	...	0.57620	0.01726	0.00588	0.16379

5 rows × 85 columns

In [57]:
```
!pip install factor_analyzer
```

```
Collecting factor_analyzer
  Downloading factor_analyzer-0.5.0.tar.gz (42 kB)
     ---------------------------------------- 42.5/42.5 kB 296.1 kB/s eta 0:00:00
  Installing build dependencies: started
  Installing build dependencies: finished with status 'done'
  Getting requirements to build wheel: started
  Getting requirements to build wheel: finished with status 'done'
  Preparing metadata (pyproject.toml): started
  Preparing metadata (pyproject.toml): finished with status 'done'
Requirement already satisfied: numpy in c:\users\s singh\anaconda3\lib\site-packages (from factor_analyzer) (1.21.5)
Requirement already satisfied: scipy in c:\users\s singh\anaconda3\lib\site-packages (from factor_analyzer) (1.9.1)
Requirement already satisfied: pandas in c:\users\s singh\anaconda3\lib\site-packages (from factor_analyzer) (1.4.4)
Requirement already satisfied: scikit-learn in c:\users\s singh\anaconda3\lib\site-packages (from factor_analyzer) (1.0.2)
Collecting pre-commit
  Downloading pre_commit-3.5.0-py2.py3-none-any.whl (203 kB)
     ---------------------------------------- 203.7/203.7 kB 1.0 MB/s eta 0:00:00
Requirement already satisfied: pytz>=2020.1 in c:\users\s singh\anaconda3\lib\site-packages (from pandas->factor_analyzer) (202
2.1)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\s singh\anaconda3\lib\site-packages (from pandas->factor_anal
yzer) (2.8.2)
Requirement already satisfied: pyyaml>=5.1 in c:\users\s singh\anaconda3\lib\site-packages (from pre-commit->factor_analyzer)
(6.0)
Collecting identify>=1.0.0
  Downloading identify-2.5.31-py2.py3-none-any.whl (98 kB)
     ---------------------------------------- 98.9/98.9 kB 5.9 MB/s eta 0:00:00
Collecting cfgv>=2.0.0
  Downloading cfgv-3.4.0-py2.py3-none-any.whl (7.2 kB)
Collecting virtualenv>=20.10.0
  Downloading virtualenv-20.24.6-py3-none-any.whl (3.8 MB)
     ---------------------------------------- 3.8/3.8 MB 2.1 MB/s eta 0:00:00
Collecting nodeenv>=0.11.1
  Downloading nodeenv-1.8.0-py2.py3-none-any.whl (22 kB)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\s singh\anaconda3\lib\site-packages (from scikit-learn->factor_
analyzer) (2.2.0)
Requirement already satisfied: joblib>=0.11 in c:\users\s singh\anaconda3\lib\site-packages (from scikit-learn->factor_analyze
r) (1.1.0)
Requirement already satisfied: setuptools in c:\users\s singh\anaconda3\lib\site-packages (from nodeenv>=0.11.1->pre-commit->fa
ctor_analyzer) (63.4.1)
Requirement already satisfied: six>=1.5 in c:\users\s singh\anaconda3\lib\site-packages (from python-dateutil>=2.8.1->pandas->f
actor_analyzer) (1.16.0)
Collecting filelock<4,>=3.12.2
  Downloading filelock-3.13.1-py3-none-any.whl (11 kB)
Collecting platformdirs<4,>=3.9.1
  Downloading platformdirs-3.11.0-py3-none-any.whl (17 kB)
Collecting distlib<1,>=0.3.7
  Downloading distlib-0.3.7-py2.py3-none-any.whl (468 kB)
     ---------------------------------------- 468.9/468.9 kB 4.9 MB/s eta 0:00:00
Building wheels for collected packages: factor_analyzer
  Building wheel for factor_analyzer (pyproject.toml): started
  Building wheel for factor_analyzer (pyproject.toml): finished with status 'done'
  Created wheel for factor_analyzer: filename=factor_analyzer-0.5.0-py2.py3-none-any.whl size=42551 sha256=08719c20ba645583d430
66ac13417e9a6e181ca71e29cd0bd414fa8f2a57e68b
  Stored in directory: c:\users\s singh\appdata\local\pip\cache\wheels\dc\d9\72\5261b2f7c80c1de8c85a0d32a8deea3879f6346ead6a85d
910
Successfully built factor_analyzer
Installing collected packages: distlib, platformdirs, nodeenv, identify, filelock, cfgv, virtualenv, pre-commit, factor_analyze
r
  Attempting uninstall: platformdirs
    Found existing installation: platformdirs 2.5.2
    Uninstalling platformdirs-2.5.2:
      Successfully uninstalled platformdirs-2.5.2
  Attempting uninstall: filelock
    Found existing installation: filelock 3.6.0
    Uninstalling filelock-3.6.0:
      Successfully uninstalled filelock-3.6.0
Successfully installed cfgv-3.4.0 distlib-0.3.7 factor_analyzer-0.5.0 filelock-3.13.1 identify-2.5.31 nodeenv-1.8.0 platformdir
s-3.11.0 pre-commit-3.5.0 virtualenv-20.24.6
```

In [60]:
```
df_train.describe().T
```

Out[60]:

	count	mean	std	min	25%	50%	75%	max
UID	27321.0	257331.996303	21343.859725	220342.0	238816.000000	257220.000000	275818.000000	294334.00000
BLOCKID	0.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN
SUMLEVEL	27321.0	140.000000	0.000000	140.0	140.000000	140.000000	140.000000	140.00000
COUNTYID	27321.0	85.646426	98.333097	1.0	29.000000	63.000000	109.000000	840.00000
STATEID	27321.0	28.271806	16.392846	1.0	13.000000	28.000000	42.000000	72.00000
...	...	...	...	...	...	...	...	...
pct_own	27053.0	0.640434	0.226640	0.0	0.502780	0.690840	0.817460	1.00000
married	27130.0	0.508300	0.136860	0.0	0.425102	0.526665	0.605760	1.00000
married_snp	27130.0	0.047537	0.037640	0.0	0.020810	0.038840	0.065100	0.71429
separated	27130.0	0.019089	0.020796	0.0	0.004530	0.013460	0.027488	0.71429
divorced	27130.0	0.100248	0.049055	0.0	0.065800	0.095205	0.129000	1.00000

74 rows × 8 columns

In [62]:
```
train.columns
```

Out[62]:
```
Index(['UID', 'SUMLEVEL', 'COUNTYID', 'STATEID', 'state', 'state_ab', 'city',
       'place', 'type', 'primary', 'zip_code', 'area_code', 'lat', 'lng',
       'ALand', 'AWater', 'pop', 'male_pop', 'female_pop', 'rent_mean',
       'rent_median', 'rent_stdev', 'rent_sample_weight', 'rent_samples',
       'rent_gt_10', 'rent_gt_15', 'rent_gt_20', 'rent_gt_25', 'rent_gt_30',
       'rent_gt_35', 'rent_gt_40', 'rent_gt_50', 'universe_samples',
       'used_samples', 'hi_mean', 'hi_median', 'hi_stdev', 'hi_sample_weight',
       'hi_samples', 'family_mean', 'family_median', 'family_stdev',
       'family_sample_weight', 'family_samples', 'hc_mortgage_mean',
       'hc_mortgage_median', 'hc_mortgage_stdev', 'hc_mortgage_sample_weight',
       'hc_mortgage_samples', 'hc_mean', 'hc_median', 'hc_stdev', 'hc_samples',
       'hc_sample_weight', 'home_equity_second_mortgage', 'second_mortgage',
       'home_equity', 'debt', 'second_mortgage_cdf', 'home_equity_cdf',
       'debt_cdf', 'hs_degree', 'hs_degree_male', 'hs_degree_female',
       'male_age_mean', 'male_age_median', 'male_age_stdev',
       'male_age_sample_weight', 'male_age_samples', 'female_age_mean',
       'female_age_median', 'female_age_stdev', 'female_age_sample_weight',
       'female_age_samples', 'pct_own', 'married', 'married_snp', 'separated',
       'divorced', 'split', 'bad_debt', 'good_debt', 'population_density',
       'median_age', 'pop_bins'],
      dtype='object')
```

In [63]:
```
train['type'].unique()
```

Out[63]:
```
array(['City', 'Urban', 'Town', 'CDP', 'Village', 'Borough'], dtype=object)
```

In [64]:
```
type_dict={'type':{'City':1, 'Urban':2, 'Town':3, 'CDP':4, 'Village':5, 'Borough':6}}
train.replace(type_dict,inplace=True)
```

In [65]:
```
test.replace(type_dict,inplace=True)
```

In [66]:
```
train['type'].unique()
```

Out[66]:
```
array([1, 2, 3, 4, 5, 6], dtype=int64)
```

In [67]:
```
test['type'].unique()
```

Out[67]:
```
array([4, 1, 6, 3, 5, 2], dtype=int64)
```

In [68]:
```
feature_cols=['COUNTYID','STATEID','zip_code','type','pop', 'family_mean','second_mortgage', 'home_equity', 'debt','hs_degree',
              'pct_own', 'married','separated', 'divorced']
```

In [69]:
```
X_train = train[feature_cols]
y_train = train['hc_mortgage_mean']
```

In [70]:
```
X_test = test[feature_cols]
y_test = test['hc_mortgage_mean']
```

In [71]:
```
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_absolute_error,mean_squared_error,accuracy_score
```

In [72]:
```
X_train.head()
```

Out[72]:

	COUNTYID	STATEID	zip_code	type	pop	family_mean	second_mortgage	home_equity	debt	hs_degree	pct_own	married	separated	divorced
0	53	36	13346	1	5230	67994.14790	0.02077	0.08919	0.52963	0.89288	0.79046	0.57851	0.01240	0.08770
1	141	18	46616	1	2633	50670.10337	0.02222	0.04274	0.60855	0.90487	0.52483	0.34886	0.01426	0.09030
2	63	18	46122	1	6881	95262.51431	0.00000	0.09512	0.73484	0.94288	0.85331	0.64745	0.01607	0.10657
3	127	72	927	2	2700	56401.68133	0.01086	0.01086	0.52714	0.91500	0.65037	0.47257	0.02021	0.10106
4	161	20	66502	1	5637	54053.42396	0.05426	0.05426	0.51938	1.00000	0.13046	0.12356	0.00000	0.03109

In [73]:
```
X_test.head()
```

Out[73]:

	COUNTYID	STATEID	zip_code	type	pop	family_mean	second_mortgage	home_equity	debt	hs_degree	pct_own	married	separated	divorced
27321	163	26	48239	4	3417	53802.87122	0.06443	0.07651	0.63624	0.91047	0.70252	0.28217	0.03813	0.14299
27322	1	23	4210	1	3796	85642.22095	0.01175	0.14375	0.64755	0.94290	0.85128	0.64221	0.00000	0.13377
27323	15	42	14871	6	3944	65694.06582	0.01316	0.06497	0.45395	0.89238	0.81897	0.59961	0.01358	0.10026
27324	231	21	42633	1	2508	44156.38709	0.00995	0.01741	0.41915	0.60908	0.84609	0.56953	0.04694	0.12489
27325	355	48	78410	3	6230	123527.02420	0.00000	0.03440	0.63188	0.86297	0.79077	0.57620	0.00588	0.16379

In [74]:
```
sc = StandardScaler()
X_train_scaled = sc.fit_transform(X_train)
X_test_scaled = sc.fit_transform(X_test)
```

In [75]:
```
lr = LinearRegression()
lr.fit(X_train_scaled, y_train)
```

Out[75]:
```
LinearRegression()
```

In [76]:
```
y_pred= lr.predict(X_test_scaled)
```

In [77]:
```
r2_score(y_test,y_pred)
```

Out[77]:
```
0.7381882934134452
```

In [78]:
```
mean_absolute_error(y_test, y_pred)
```

Out[78]:
```
233.86965694140082
```

In [79]:
```
mean_squared_error(y_test, y_pred)
```

Out[79]:
```
103818.4048673347
```

In [80]:
```
np.sqrt(mean_squared_error(y_test,y_pred))
```

Out[80]:
```
322.2086356188094
```

In [81]:
```
r2_score(y_train, lr.predict(X_train_scaled))
```

Out[81]:
```
0.734344756627955
```

In [82]:
```
lr.coef_
```

Out[82]:
```
array([ -28.50842455,  -21.7100607 ,  -22.98370175,  -57.43101333,
         -4.78426374,  558.7402445 ,   -0.55955638,   70.89657588,
         12.81271881, -113.18431746, -176.51983734,    8.10645154,
          5.24214879,  -55.79637445])
```

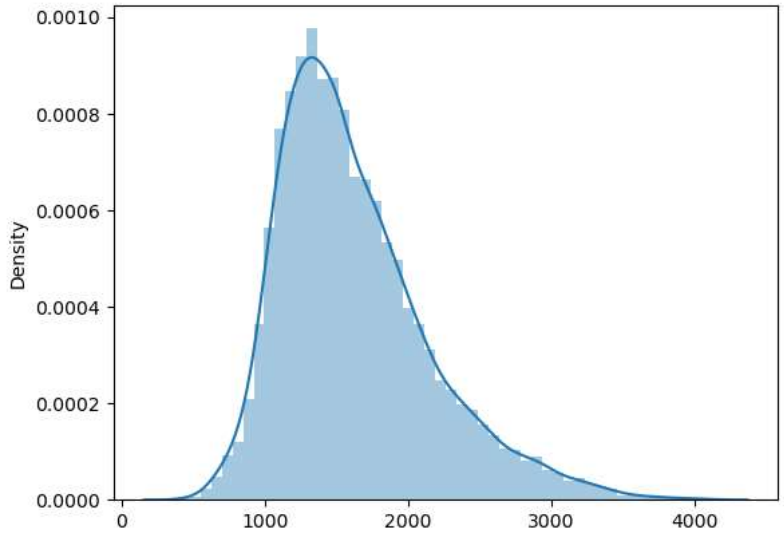
In [83]:
```
X_train.columns
```

Out[83]:
```
Index(['COUNTYID', 'STATEID', 'zip_code', 'type', 'pop', 'family_mean',
       'second_mortgage', 'home_equity', 'debt', 'hs_degree', 'pct_own',
       'married', 'separated', 'divorced'],
      dtype='object')
```

In [84]:
```
state = train['STATEID'].unique()
state
```

Out[84]:
```
array([36, 18, 72, 20,  1, 48, 45,  6,  5, 24, 17, 19, 47, 32, 22,  8, 44,
       28, 34, 41,  4, 12, 55, 42, 37, 51, 26, 39, 40, 13, 16, 46, 27, 29,
       53, 56,  9, 54, 21, 25, 11, 15, 30,  2, 33, 49, 50, 31, 38, 35, 23,
       10], dtype=int64)
```

In [85]:
```
sns.distplot(y_pred)
plt.show()
```



In [ ]:
```

```