```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler, LabelEncoder
import datetime
```

```
pip install pandas numpy matplotlib seaborn scikit-learn
```

```
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (1.5.3)
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (1.25.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.7.1)
Requirement already satisfied: seaborn in /usr/local/lib/python3.10/dist-packages (0.13.1)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.2.2)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2023.4)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.2.
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (4.4
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (23.2)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (9.4.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (3.1.
Requirement already satisfied: scipy>=1.3.2 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.11.4
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.3.2
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.1->p
```

```
df_train=pd.read_csv('train.csv')
df_test=pd.read_csv('test.csv')
```

```
df_train.head()
```

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

5 rows × 80 columns

```
df_test.head()
```

	UID	BLOCKID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	
0	255504	NaN	140	163	26	Michigan	MI	Detroit	De H
1	252676	NaN	140	1	23	Maine	ME	Auburn	/
2	276314	NaN	140	15	42	Pennsylvania	PA	Pine City	M
3	248614	NaN	140	231	21	Kentucky	KY	Monticello	Mo
4	286865	NaN	140	355	48	Texas	TX	Corpus Christi	

5 rows × 80 columns

```
df_train.shape
```

```
(27321, 80)
```

```
df_test.shape
```

```
(11709, 80)
```

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

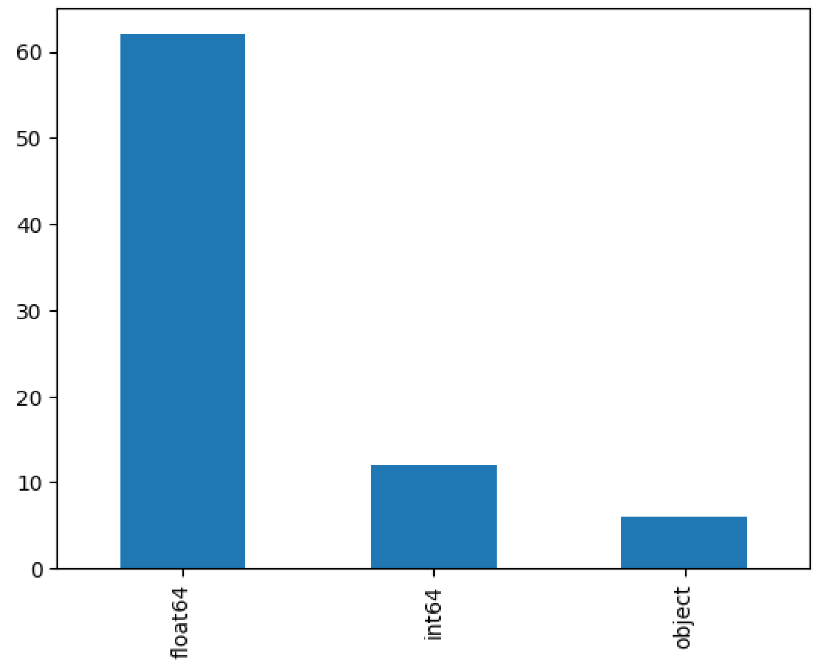
```
123
```

```
df_train.dtypes
```

```
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
```

```
df_train.dtypes.value_counts().plot(kind='bar')
```

<Axes: >




```
df_train.describe(include='O')
```

	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

```
#This flag will help us split the data back later
df_train['split']= 'Train'
df_test['split']= 'Test'
```

```
df_combined=df_train.append(df_test, ignore_index=True)
df_combined.head()
```

```
<ipython-input-15-bb661d45caa3>:1: FutureWarning: The frame.append method is depreca
  df_combined=df_train.append(df_test, ignore_index=True)
```

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

5 rows × 81 columns

```
df_combined.tail()
```

	UID	BLOCKID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city
39025	238088	NaN	140	105	12	Florida	FL	Lakelanc
39026	242811	NaN	140	31	17	Illinois	IL	Chicagc
39027	250127	NaN	140	9	25	Massachusetts	MA	Lawrence
39028	241096	NaN	140	27	19	Iowa	IA	Carrol
39029	287763	NaN	140	453	48	Texas	TX	Austir

5 rows × 81 columns

```
df_combined.shape
```

```
(39030, 81)
```

```
df_combined.isna().sum()
```

```
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
```

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

```
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
```

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

```
df_combined.isna().sum()/len(df_combined)*100
```

```
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
```

```
# 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
```

```
['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']
```

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

```
rent_mean
rent_median
rent_stdev
rent_sample_weight
rent_samples
```

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

```
df_combined.isna().sum()/len(df_combined)*100
```

```
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
```

```
df_combined.shape
```

```
(39030, 80)
```

```
# Drop duplicate observations
df_combined.drop_duplicates(inplace=True)
df_combined.shape
```

```
(38838, 80)
```

```
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)
```

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

	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

```
!pip install geopandas
import warnings
warnings.filterwarnings('ignore')
```

```
Requirement already satisfied: geopandas in /usr/local/lib/python3.10/dist-packages (0.13.2)
Requirement already satisfied: fiona>=1.8.19 in /usr/local/lib/python3.10/dist-packages (from geopandas) (1.9.5)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from geopandas) (23.2)
Requirement already satisfied: pandas>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from geopandas) (1.5.3)
Requirement already satisfied: pyproj>=3.0.1 in /usr/local/lib/python3.10/dist-packages (from geopandas) (3.6.1)
Requirement already satisfied: shapely>=1.7.1 in /usr/local/lib/python3.10/dist-packages (from geopandas) (2.0.2)
Requirement already satisfied: attrs>=19.2.0 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopa
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopandas)
Requirement already satisfied: click~=8.0 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopanda
Requirement already satisfied: click-plugins>=1.0 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->
Requirement already satisfied: cligj>=0.5 in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopanda
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopandas) (1.1
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/dist-packages (from fiona>=1.8.19->geopanda
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1.0->geopan
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1.0->geopa
```

```
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
```

	state	city	state_ab	place	lat	lng	geome
11980	Massachusetts	Worcester	MA	Worcester City	42.254262	-71.800347	PC (-71.8C 42.25
26018	New York	Corona	NY	Harbor Hills	40.751809	-73.853582	PC (-73.85 40.75
7829	Maryland	Glen Burnie	MD	Glen Burnie	39.127273	-76.635265	PC (-76.63 39.12
2077	Florida	Tampa	FL	Egypt Lake-leto	28.029063	-82.495395	PC (-82.4 28.02
1701	Illinois	Chicago	IL	Lincolnwood	41.967289	-87.652434	PC (-87.65 41.96
...	...	...	...	...	...	...	

```
#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_secon
df_combined.head()
```

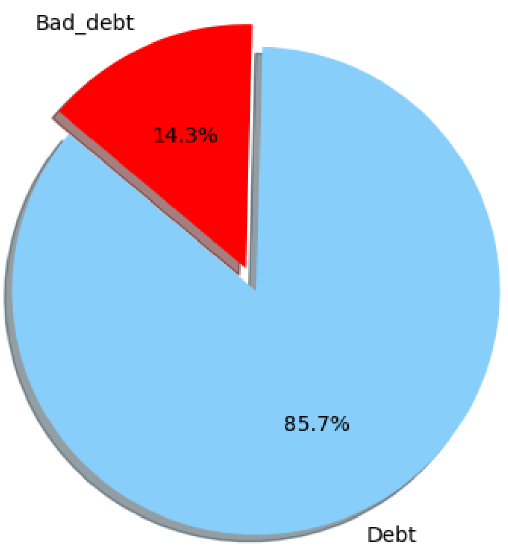
	UID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	type
0	267822	140	53	36	New York	NY	Hamilton	Hamilton	City
1	246444	140	141	18	Indiana	IN	South Bend	Roseland	City
2	245683	140	63	18	Indiana	IN	Danville	Danville	City
3	279653	140	127	72	Puerto Rico	PR	San Juan	Guaynabo	Urban
4	247218	140	161	20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

5 rows × 81 columns

```
import matplotlib.pyplot as plt
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()
```




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

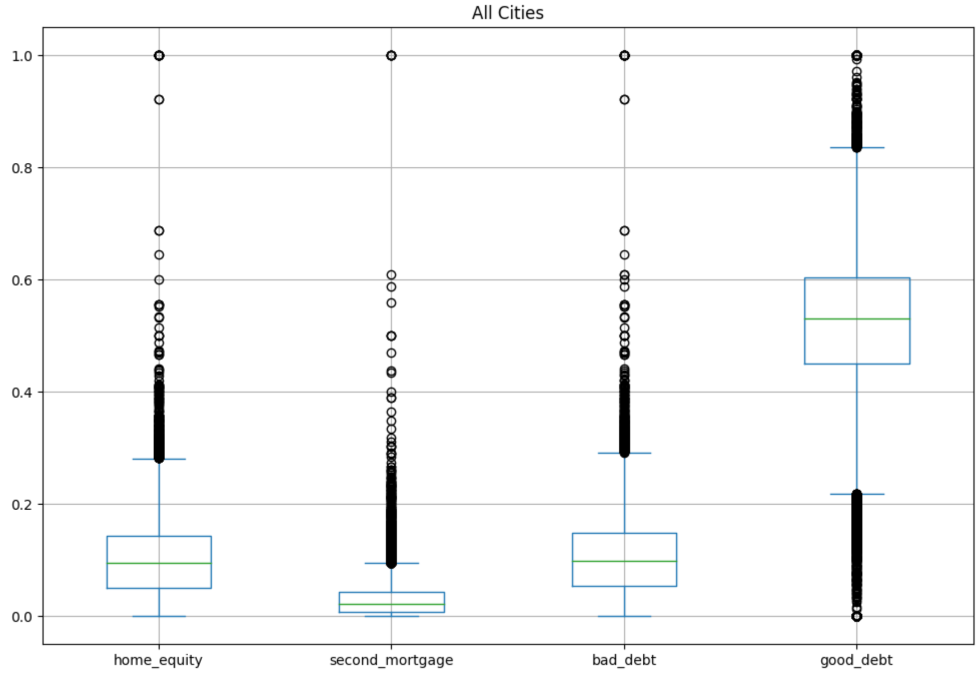
	UID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	type
0	267822	140	53	36	New York	NY	Hamilton	Hamilton	City
1	246444	140	141	18	Indiana	IN	South Bend	Roseland	City
2	245683	140	63	18	Indiana	IN	Danville	Danville	City
3	279653	140	127	72	Puerto Rico	PR	San Juan	Guaynabo	Urban
4	247218	140	161	20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

5 rows × 82 columns

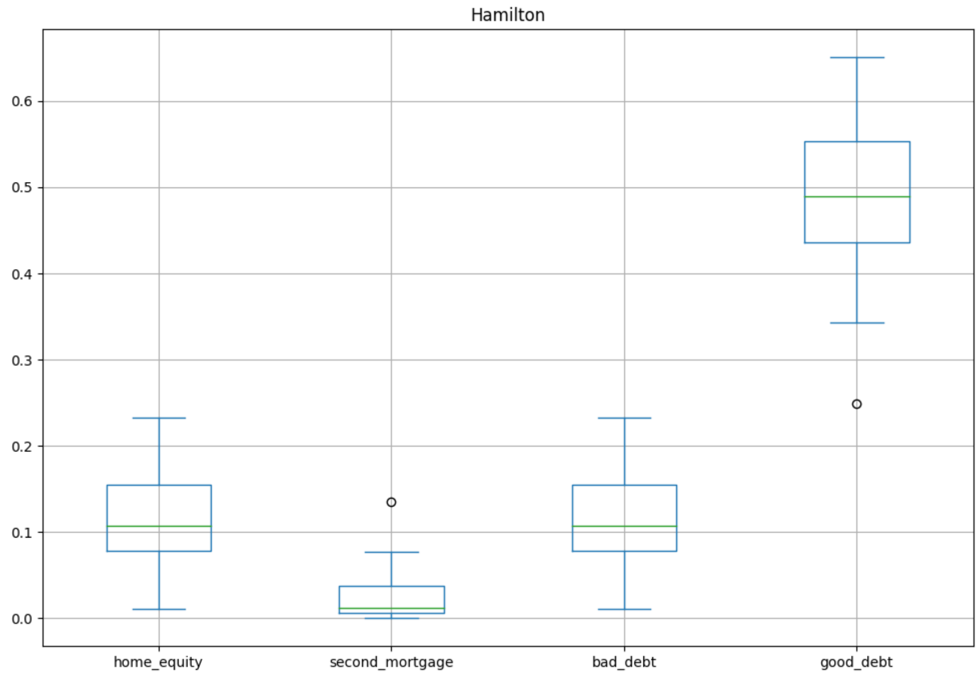
```
df_combined.columns
```

```
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')
```

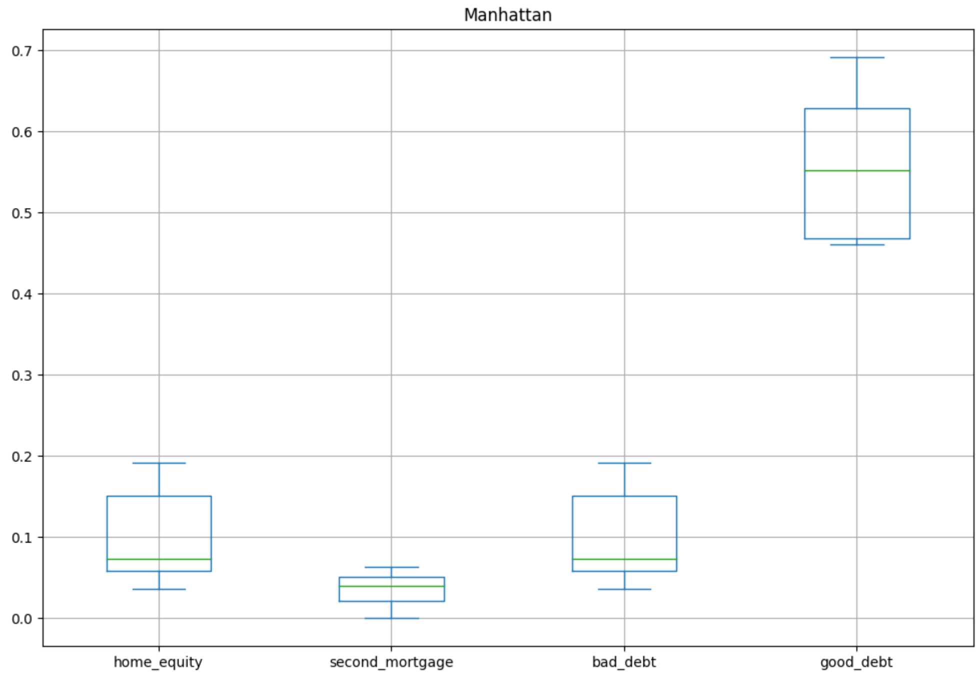
```
all_cities = df_combined[['home_equity','second_mortgage','bad_debt', 'good_debt']]
all_cities.plot.box(figsize=(12,8),grid=True)
plt.title('All Cities')
plt.show()
```




```
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()
```

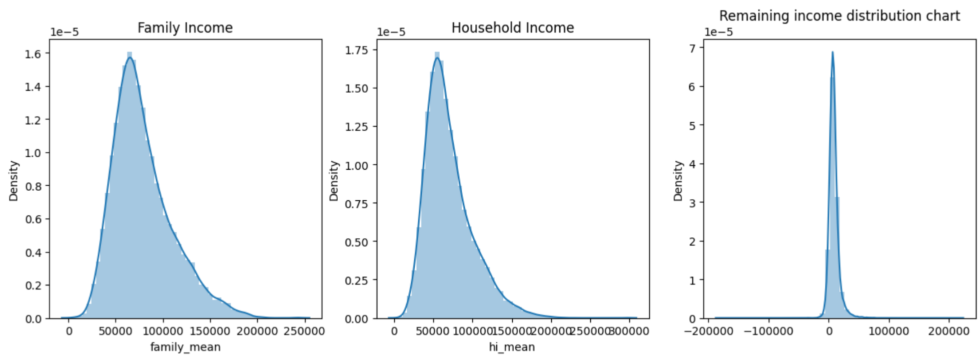



```
Manhattan = df_combined[df_combined['city']=='Manhattan']
Manhattan = Manhattan[['home_equity','second_mortgage','bad_debt', 'good_debt']]
Manhattan.plot.box(figsize=(12,8),grid=True)
plt.title('Manhattan')
plt.show()
```




```
import seaborn as sns
plt.figure(figsize=(15,10))

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()
```




```
df_combined['population_density'] = df_combined['pop']/df_combined['ALand']
```

```
df_combined.head()
```

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

5 rows × 83 columns

```
# Weighted average
# median_age=((male_age_median * male_pop)+(female_age_median*female_pop))/(male_pop+female_pop)
#           =((40*10)+(50*30))/40
#           =(400+1500)/40
#           =190/4
#           =47.5
df_combined['median_age']=((df_combined['male_age_median'] * df_combined['male_pop'])+(df_combined['female_age_median'
```

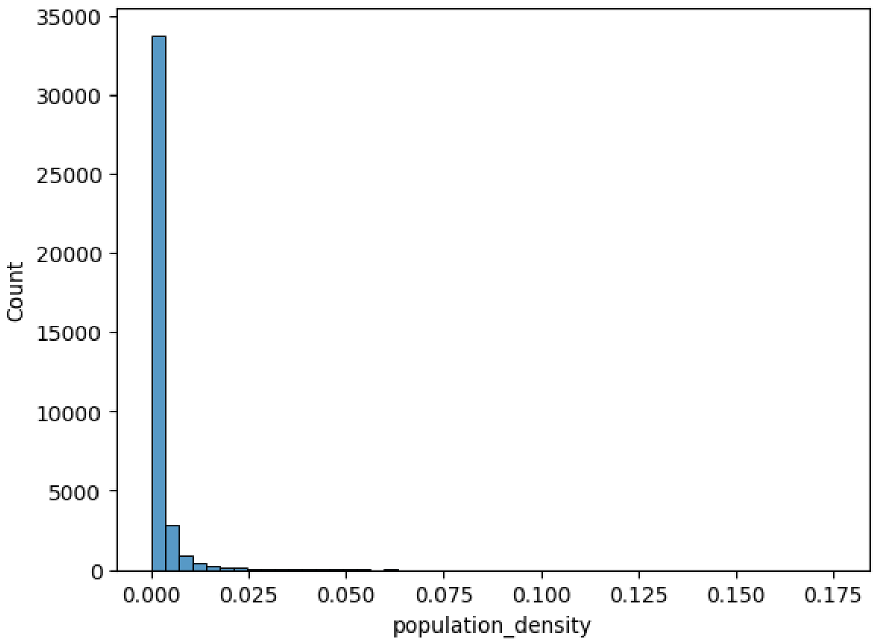
```
df_combined.head()
```

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

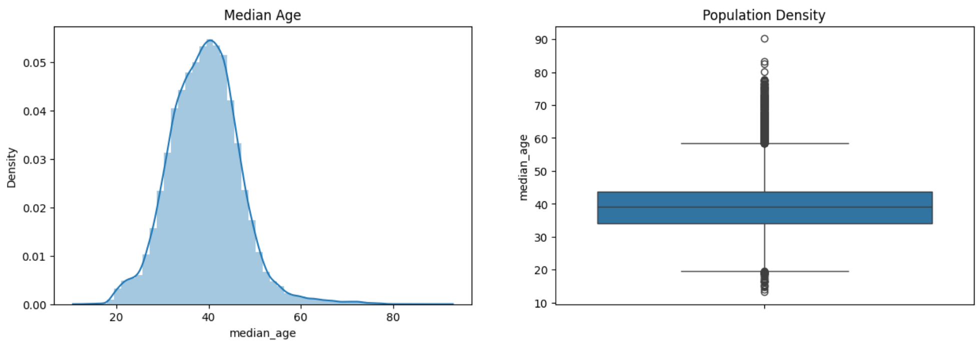
5 rows × 84 columns

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

```
<Axes: xlabel='population_density', ylabel='Count'>
```




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




```
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()
```

```
very low     38472
low            348
medium          12
high             5
very high        1
Name: pop_bins, dtype: int64
```

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

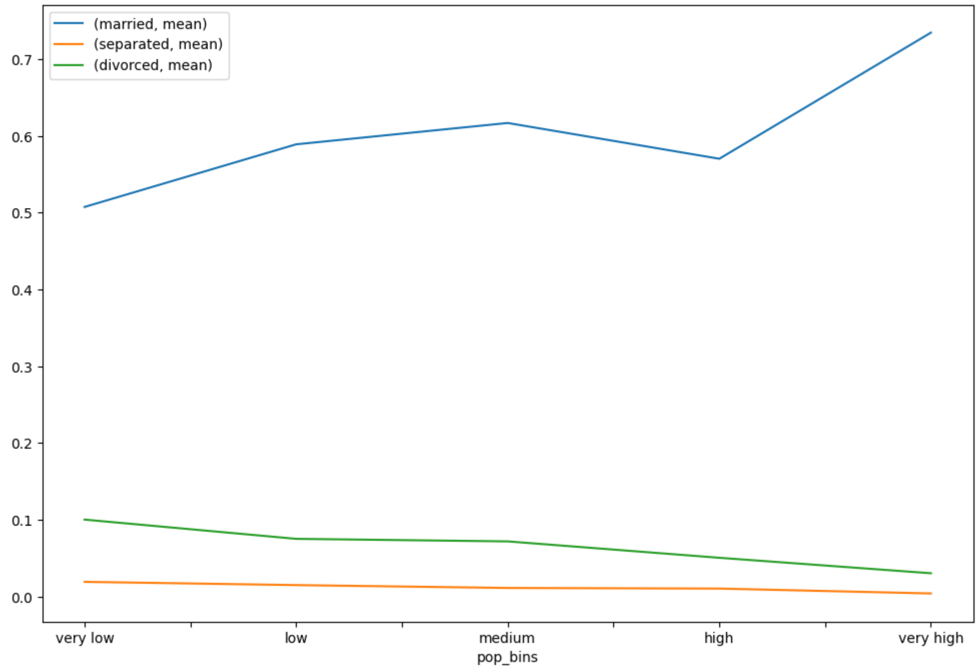
	married	separated	divorced
pop_bins			
very low	38472	38472	38472
low	348	348	348
medium	12	12	12
high	5	5	5
very high	1	1	1

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

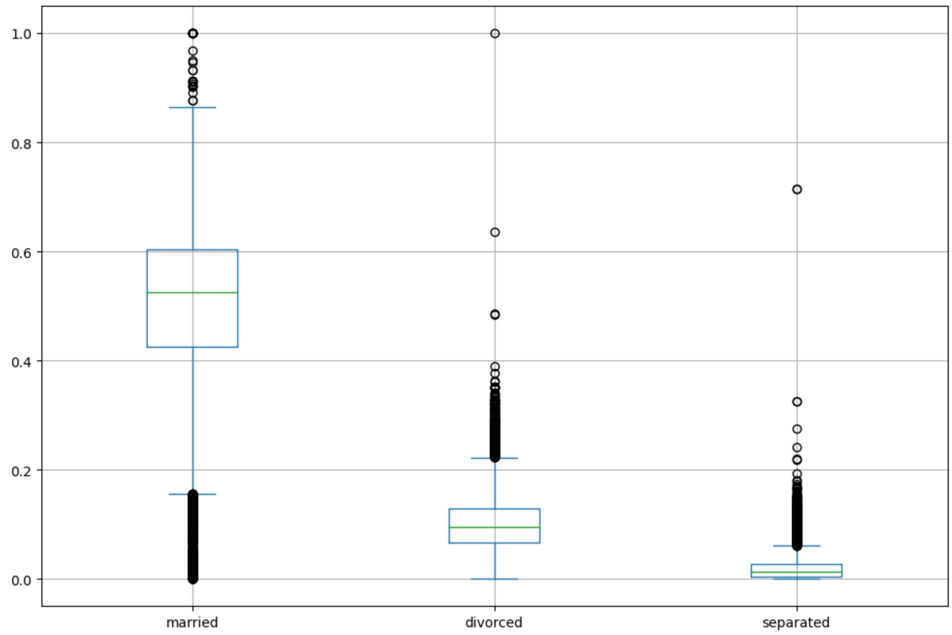
	married		separated		divorced	
	mean	median	mean	median	mean	median
pop_bins						
very low	0.507647	0.526210	0.019163	0.013580	0.100263	0.094965
low	0.589247	0.601815	0.014929	0.010255	0.075192	0.069340
medium	0.617047	0.605765	0.011203	0.007745	0.071870	0.069090
high	0.570438	0.614130	0.010398	0.005520	0.050514	0.056690
very high	0.734740	0.734740	0.004050	0.004050	0.030360	0.030360

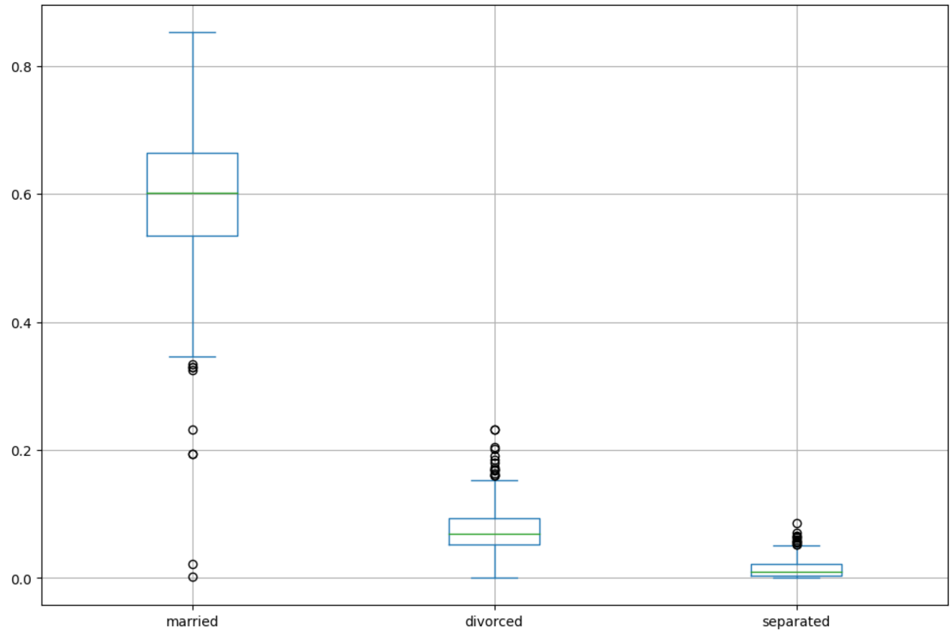
```
plt.figure(figsize=(12,8))
pop_bin_married=df_combined.groupby(by='pop_bins')[['married','separated','divorced']].agg(["mean"])
pop_bin_married.plot(figsize=(12,8))
plt.legend(loc='best')
plt.show()
```

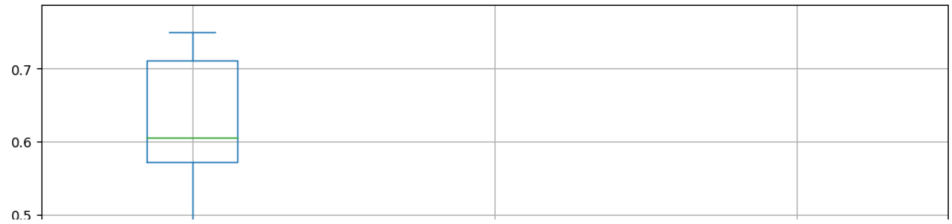
```
<Figure size 1200x800 with 0 Axes>
```



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


1.0
0.8
0.6
0.4
0.2
0.0
married
divorced
separated


0.8
0.6
0.4
0.2
0.0
married
divorced
separated


0.7
0.6
0.5

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

	mean
state	
Alabama	765.872557
Alaska	1190.093590
Arizona	1084.462392
Arkansas	716.544987
California	1465.019694

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

	mean
state	
Alabama	65311.510962
Alaska	91911.137520
Arizona	73020.627940
Arkansas	64234.705963
California	87599.537172

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

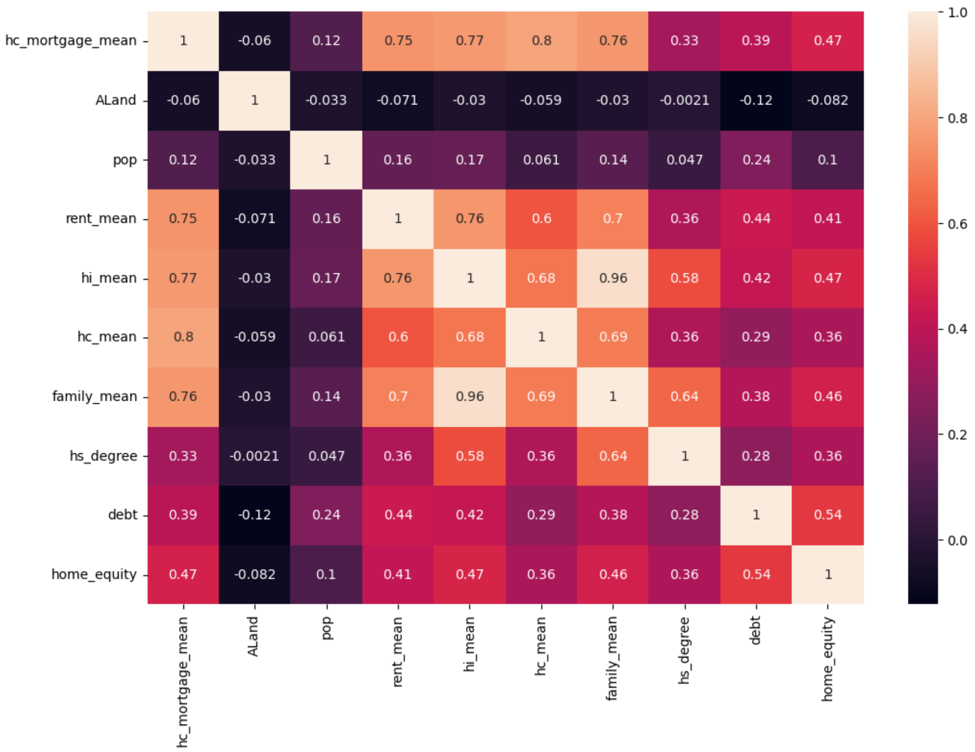
```
state
Alabama                 1.172646
Alaska                  1.294831
Arizona                 1.485145
Arkansas                1.115511
California              1.672406
Colorado                1.362639
Connecticut             1.272709
Delaware                1.311538
District of Columbia    1.357102
Florida                 1.576506
Name: mean, dtype: float64
```

```
sum(df_combined['rent_mean'])/sum(df_combined['family_mean'])
```

```
0.013360285332548792
```

```
plt.figure(figsize=(12,8))
```

```
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()
```




```
train = df_combined[df_combined['split'] == 'Train']
test = df_combined[df_combined['split'] == 'Test']
```

```
train.head()
```

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

5 rows × 85 columns

```
test.head()
```