Real Estate

A banking institution requires actionable insights from the perspective of Mortgage-Backed Securities, Geographic Business Investment and Real Estate Analysis.

The objective is to identify white spaces/potential business in the mortgage loan. The mortgage bank would like to identify potential monthly mortgage expenses for each of region based on factors which are primarily monthly family income in a region and rented value of the real estate. Some of the regions are growing rapidly and Competitor banks are selling mortgage loans to subprime customers at a lower interest rate. The bank is strategizing for better market penetration and targeting new customers. A statistical model needs to be created to predict the potential demand in dollars amount of loan for each of the region in the USA. Also, there is a need to create a dashboard which would refresh periodically post data retrieval from the agencies. This would help to monitor the key metrics and trends.

Import the required libraries

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

Import the data

In [2]:
```
df = pd.read_csv('T:\Masters In Data Science\Capstone Project\Project 1\\train.csv')
```

In [3]:
```
df.head()
```

Out[3]:

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

5 rows × 80 columns

In [4]:
```
df.shape
```

Out[4]:
```
(27321, 80)
```

In [5]:
```
df.info()   ## checking for null values in the data as well as data types of several v
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27321 entries, 0 to 27320
Data columns (total 80 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   UID                          27321 non-null  int64  
 1   BLOCKID                      0 non-null      float64
 2   SUMLEVEL                     27321 non-null  int64  
 3   COUNTYID                     27321 non-null  int64  
 4   STATEID                      27321 non-null  int64  
 5   state                        27321 non-null  object 
 6   state_ab                     27321 non-null  object 
 7   city                         27321 non-null  object 
 8   place                        27321 non-null  object 
 9   type                         27321 non-null  object 
 10  primary                      27321 non-null  object 
 11  zip_code                     27321 non-null  int64  
 12  area_code                    27321 non-null  int64  
 13  lat                          27321 non-null  float64
 14  lng                          27321 non-null  float64
 15  ALand                        27321 non-null  float64
 16  AWater                       27321 non-null  int64  
 17  pop                          27321 non-null  int64  
 18  male_pop                     27321 non-null  int64  
 19  female_pop                   27321 non-null  int64  
 20  rent_mean                    27007 non-null  float64
 21  rent_median                  27007 non-null  float64
 22  rent_stdev                   27007 non-null  float64
 23  rent_sample_weight           27007 non-null  float64
 24  rent_samples                 27007 non-null  float64
 25  rent_gt_10                   27007 non-null  float64
 26  rent_gt_15                   27007 non-null  float64
 27  rent_gt_20                   27007 non-null  float64
 28  rent_gt_25                   27007 non-null  float64
 29  rent_gt_30                   27007 non-null  float64
 30  rent_gt_35                   27007 non-null  float64
 31  rent_gt_40                   27007 non-null  float64
 32  rent_gt_50                   27007 non-null  float64
 33  universe_samples             27321 non-null  int64  
 34  used_samples                 27321 non-null  int64  
 35  hi_mean                      27053 non-null  float64
 36  hi_median                    27053 non-null  float64
 37  hi_stdev                     27053 non-null  float64
 38  hi_sample_weight             27053 non-null  float64
 39  hi_samples                   27053 non-null  float64
 40  family_mean                  27023 non-null  float64
 41  family_median                27023 non-null  float64
 42  family_stdev                 27023 non-null  float64
 43  family_sample_weight         27023 non-null  float64
 44  family_samples               27023 non-null  float64
 45  hc_mortgage_mean             26748 non-null  float64
 46  hc_mortgage_median           26748 non-null  float64
 47  hc_mortgage_stdev            26748 non-null  float64
 48  hc_mortgage_sample_weight    26748 non-null  float64
 49  hc_mortgage_samples          26748 non-null  float64
 50  hc_mean                      26721 non-null  float64
 51  hc_median                    26721 non-null  float64
 52  hc_stdev                     26721 non-null  float64
 53  hc_samples                   26721 non-null  float64
 54  hc_sample_weight             26721 non-null  float64
 55  home_equity_second_mortgage  26864 non-null  float64
 56  second_mortgage              26864 non-null  float64
 57  home_equity                  26864 non-null  float64
 58  debt                         26864 non-null  float64
 59  second_mortgage_cdf          26864 non-null  float64
 60  home_equity_cdf              26864 non-null  float64
 61  debt_cdf                     26864 non-null  float64
 62  hs_degree                    27131 non-null  float64
 63  hs_degree_male               27121 non-null  float64
 64  hs_degree_female             27098 non-null  float64
 65  male_age_mean                27132 non-null  float64
 66  male_age_median              27132 non-null  float64
 67  male_age_stdev               27132 non-null  float64
 68  male_age_sample_weight       27132 non-null  float64
 69  male_age_samples             27132 non-null  float64
 70  female_age_mean              27115 non-null  float64
 71  female_age_median            27115 non-null  float64
 72  female_age_stdev             27115 non-null  float64
```

```
 73   female_age_sample_weight     27115 non-null   float64
 74   female_age_samples           27115 non-null   float64
 75   pct_own                      27053 non-null   float64
 76   married                      27130 non-null   float64
 77   married_snp                  27130 non-null   float64
 78   separated                    27130 non-null   float64
 79   divorced                     27130 non-null   float64
dtypes: float64(62), int64(12), object(6)
memory usage: 16.7+ MB
```

Null values treatment

In [6]:
```
df.isnull().sum()
```

Out[6]:
```
UID                0
BLOCKID        27321
SUMLEVEL           0
COUNTYID           0
STATEID            0
               ...  
pct_own          268
married          191
married_snp      191
separated        191
divorced         191
Length: 80, dtype: int64
```

In [7]:
```
df_train = df.drop('BLOCKID',axis=1)
```

In [8]:
```
df_train.head()
```

Out[8]:

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

5 rows × 79 columns

In [9]:
```
df_train.isnull().sum()
```

Out[9]:
```
UID              0
SUMLEVEL         0
COUNTYID         0
STATEID          0
state            0
              ... 
pct_own        268
married        191
married_snp    191
separated      191
divorced       191
Length: 79, dtype: int64
```

In [10]:
```
df_train.dropna(inplace=True)    ## Dropping the null values
```

In [11]:
```
df_train.isnull().sum()
```

Out[11]:
```
UID            0
SUMLEVEL       0
COUNTYID       0
STATEID        0
state          0
              ..
pct_own        0
married        0
married_snp    0
separated      0
divorced       0
Length: 79, dtype: int64
```

We are taking the top 2500 locations where Second Mortgage is highest and Percentage Ownership is also above 10%

In [12]:
```
df_train1 = df_train.nlargest(2500,['second_mortgage','pct_own'])
```

In [13]:
```
df_train1.shape
```

Out[13]:
```
(2500, 79)
```

In [14]:
```
df_train1.head()
```

Out[14]:

	UID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	
14014	264403	140	31	34	New Jersey	NJ	Passaic	Garfield City	
3285	289712	140	147	51	Virginia	VA	Farmville	Farmville	
21706	222830	140	13	4	Arizona	AZ	Scottsdale	Tempe City	
11980	251185	140	27	25	Massachusetts	MA	Worcester	Worcester City	
12896	278178	140	101	42	Pennsylvania	PA	Philadelphia	Millbourne	Bo

5 rows × 79 columns

In [15]:
```
df_train1['Bad_debt'] = df_train1['second_mortgage']+df_train1['home_equity']-df_trair
```

In [16]:
```
df_train1.head()
```

Out[16]:

	UID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	
14014	264403	140	31	34	New Jersey	NJ	Passaic	Garfield City	
3285	289712	140	147	51	Virginia	VA	Farmville	Farmville	
21706	222830	140	13	4	Arizona	AZ	Scottsdale	Tempe City	
11980	251185	140	27	25	Massachusetts	MA	Worcester	Worcester City	
12896	278178	140	101	42	Pennsylvania	PA	Philadelphia	Millbourne	Bo

5 rows × 80 columns

In [17]:
```
df_train1['Good_debt'] = df_train1['debt']-df_train1['Bad_debt']
```

In [18]:
```
df_train1.head()
```

Out[18]:

	UID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	
14014	264403	140	31	34	New Jersey	NJ	Passaic	Garfield City	
3285	289712	140	147	51	Virginia	VA	Farmville	Farmville	
21706	222830	140	13	4	Arizona	AZ	Scottsdale	Tempe City	
11980	251185	140	27	25	Massachusetts	MA	Worcester	Worcester City	
12896	278178	140	101	42	Pennsylvania	PA	Philadelphia	Millbourne	Bo

5 rows × 81 columns

In [19]:
```
df_train1.describe()
```

Out[19]:

	UID	SUMLEVEL	COUNTYID	STATEID	zip_code	area_code	lat
count	2500.000000	2500.0	2500.000000	2500.000000	2500.000000	2500.000000	2500.000000
mean	251192.994400	140.0	75.166400	23.252400	55915.810400	602.945600	37.862908
std	21841.694936	0.0	100.094679	16.302485	31729.029582	228.898513	4.785496
min	220366.000000	140.0	1.000000	1.000000	951.000000	201.000000	18.384790
25%	232083.250000	140.0	31.000000	8.000000	29280.500000	407.000000	34.055419
50%	246485.000000	140.0	53.000000	19.000000	55337.000000	626.000000	38.831048
75%	269242.250000	140.0	89.000000	37.000000	90056.000000	775.000000	41.129508
max	294317.000000	140.0	820.000000	72.000000	99701.000000	989.000000	64.851287

8 rows × 75 columns

In [20]:
```
piechart = df_train1[['place','debt','Bad_debt','Good_debt']].reset_index()
```

In [21]:
```
piechart.head()
```

Out[21]:

	index	place	debt	Bad_debt	Good_debt
0	14014	Garfield City	0.60870	0.60870	0.00000
1	3285	Farmville	0.50000	0.50000	0.00000
2	21706	Tempe City	0.54688	0.43750	0.10938
3	11980	Worcester City	0.84956	0.43363	0.41593
4	12896	Millbourne	0.93902	0.60975	0.32927

In [22]:
```
l1 = list(piechart['Bad_debt'])
l1[:10]
```

Out[22]:
```
[0.6087,
 0.5,
 0.4375,
 0.43363,
 0.60975,
 0.36364,
 0.34783,
 0.33333,
 0.40340999999999994,
 0.40984]
```

In [23]:
```
l2 = list(piechart['Good_debt'])
l2[:10]
```

Out[23]:
```
[0.0,
 0.0,
 0.10938000000000003,
 0.41592999999999997,
 0.32926999999999995,
 0.39394,
 0.34782,
 0.36110999999999993,
 0.38068,
 0.39344]
```

In [24]:
```
l3 = sum(zip(l1,l2+[0]),())
l3[:20]
```

Out[24]:
```
(0.6087,
 0.0,
 0.5,
 0.0,
 0.4375,
 0.10938000000000003,
 0.43363,
 0.41592999999999997,
 0.60975,
 0.32926999999999995,
 0.36364,
 0.39394,
 0.34783,
 0.34782,
 0.33333,
 0.36110999999999993,
 0.40340999999999994,
 0.38068,
 0.40984,
 0.39344)
```

In [25]:
```
debt_good_bad = l3[:20]

size = 10
labels_D = ['GD', 'BD'] * size
labels_D = tuple(labels_D)
labels_D
```

Out[25]:
```
('GD',
 'BD',
 'GD',
 'BD',
 'GD',
 'BD',
 'GD',
 'BD',
 'GD',
 'BD',
 'GD',
 'BD',
 'GD',
 'BD',
 'GD',
 'BD',
 'GD',
 'BD',
 'GD',
 'BD')
```

In [26]:
```
color_pal = plt.rcParams['axes.prop_cycle'].by_key()['color']
```

In [27]:
```
sns.set_style("whitegrid")

plt.figure(figsize = (10,10))

plt.pie(piechart.debt[:10], labels=piechart.place[:10],autopct = '%0.2f%%',radius=25,s
plt.pie(debt_good_bad[:20],labels =labels_D ,autopct = '%0.2f%%',radius=20,startangle
center_circle = plt.Circle((0,0),10,color='black', fc='white',linewidth=0)
fig = plt.gcf()
fig.gca().add_artist(center_circle)
plt.axis('equal')
plt.title('Debt Analysis',fontsize=20)
plt.tight_layout()
plt.show()
```

Pie chart shows the Overall debt and Good debt and Bad debt as part of overall debt for top 10 cities

Here we can see that Millbourne is having maximum debt percentage out of top 10 cities and 8.49% of the debt is good debt for the city

In [28]:
```
city_list = df_train1['city'].value_counts()[:30].index
```

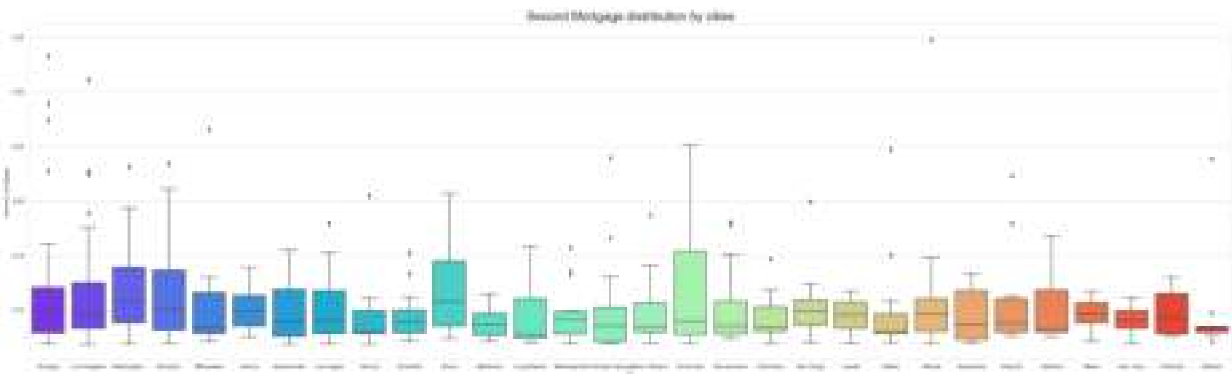
In [29]:
```
city_list
```

Out[29]:
```
Index(['Chicago', 'Los Angeles', 'Washington', 'Brooklyn', 'Milwaukee',
       'Aurora', 'Jacksonville', 'Las Vegas', 'Denver', 'Charlotte', 'Bronx',
       'Baltimore', 'Long Beach', 'Minneapolis', 'Colorado Springs',
       'New Orleans', 'Cincinnati', 'Sacramento', 'Columbus', 'San Diego',
       'Lowell', 'Dallas', 'Atlanta', 'Alexandria', 'Orlando', 'Oakland',
       'Miami', 'San Jose', 'Portland', 'Littleton'],
      dtype='object')
```

In [30]:
```
boxplot = df_train1[df_train1['city'].isin(city_list)]
```

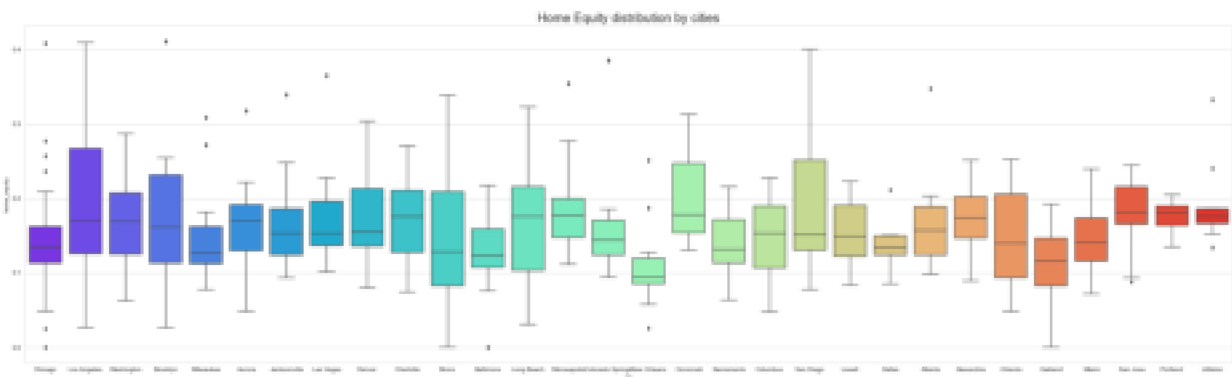
In [31]:
```
sns.set_style('whitegrid')

plt.figure(figsize = (35,10))
sns.boxplot(x='city',y='second_mortgage',data=boxplot,palette='rainbow',order=['Chicag
plt.title('Second Mortgage distribution by cities',fontsize=20)
plt.show()
```



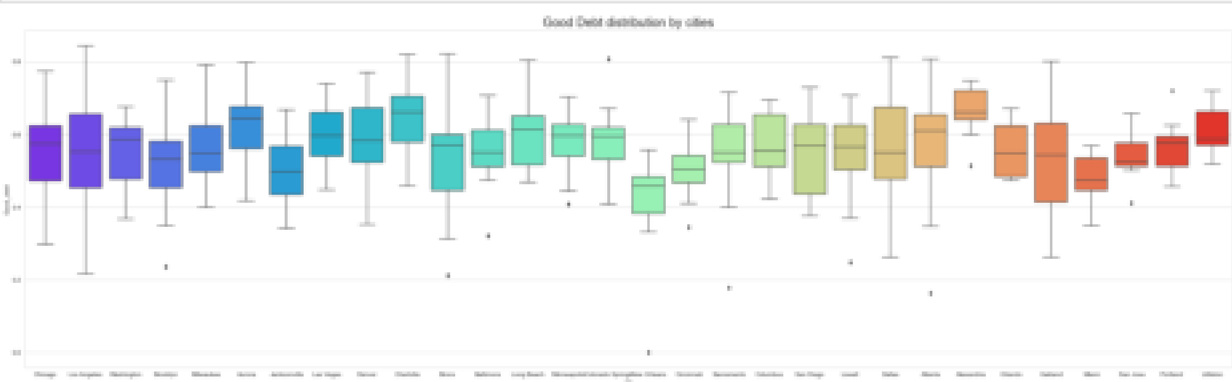
In [32]:
```
sns.set_style('whitegrid')

plt.figure(figsize = (35,10))
sns.boxplot(x='city',y='home_equity',data=boxplot,palette='rainbow',order=['Chicago',
plt.title('Home Equity distribution by cities',fontsize=20)
plt.show()
```



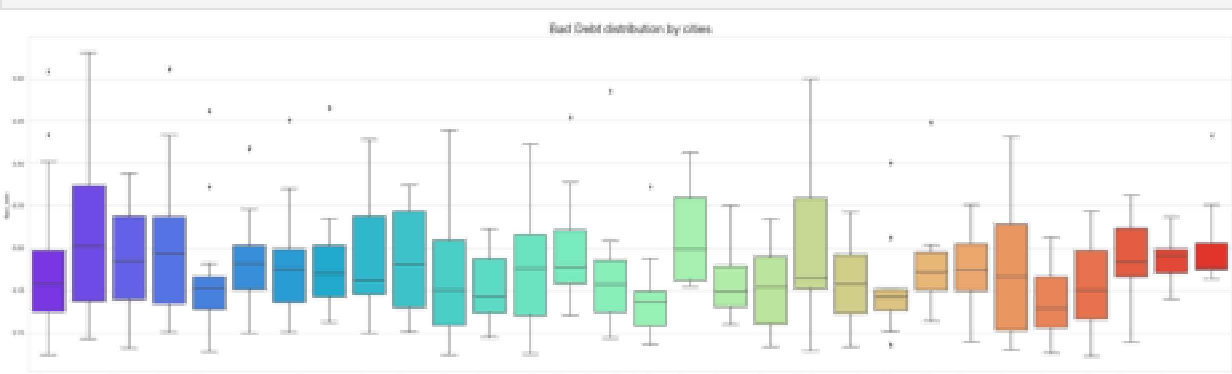
In [33]:
```
sns.set_style('whitegrid')

plt.figure(figsize = (35,10))
sns.boxplot(x='city',y='Good_debt',data=boxplot,palette='rainbow',order=['Chicago', 'L
plt.title('Good Debt distribution by cities',fontsize=20)
plt.show()
```



In [34]:
```
sns.set_style('whitegrid')

plt.figure(figsize = (35,10))
sns.boxplot(x='city',y='Bad_debt',data=boxplot,palette='rainbow',order=['Chicago', 'Lo
plt.title('Bad Debt distribution by cities',fontsize=20)
plt.show()
```

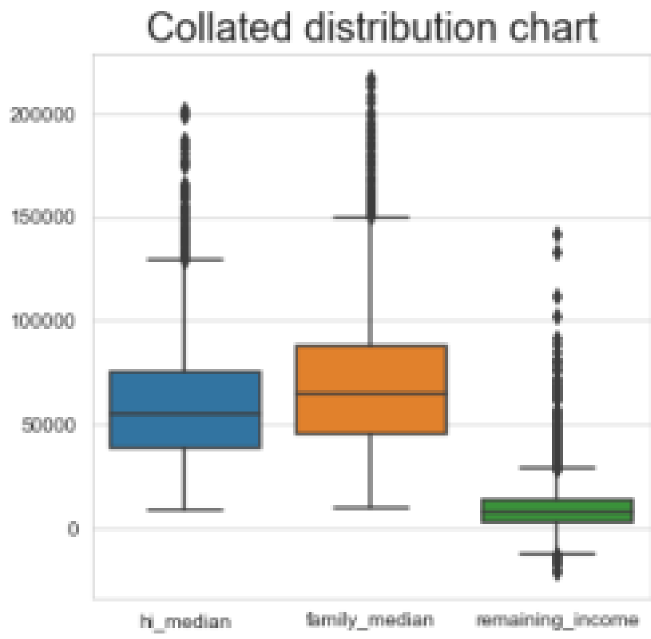


In [35]:
```
df_train1['remaining_income'] = df_train1['family_median']-df_train1['hi_median']
```

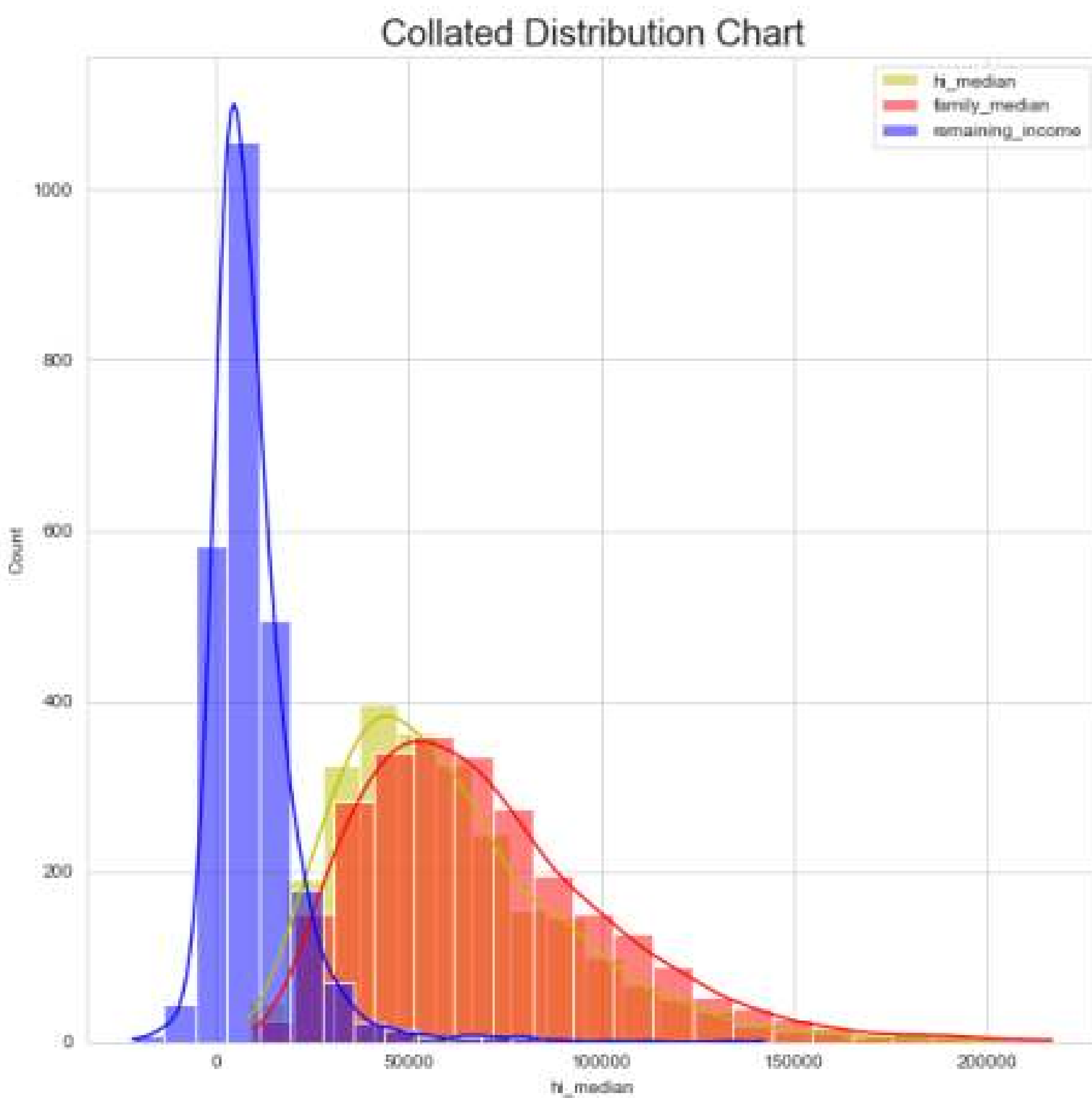
In [36]:
```
sns.set_style('whitegrid')

plt.figure(figsize = (5,5))
sns.boxplot(data=df_train1[['hi_median','family_median','remaining_income']],palette=c
```

```
plt.title('Collated distribution chart',fontsize=20)
plt.show()
```



In [37]:
```
plt.figure(figsize=(10,10))
sns.histplot(df_train1.hi_median,kde=True,bins=20,color='y',label='hi_median')
sns.histplot(df_train1.family_median,kde=True,bins=20,color='r',label='family_median')
sns.histplot(df_train1.remaining_income,kde=True,bins=20,color='b',label='remaining_in
plt.legend()
plt.title('Collated Distribution Chart',fontsize=20)
plt.show()
```



In [38]:
```
df_train1['Population_density'] = df_train1['pop'] / df_train1['ALand']
```

In [39]:
```
df_train1.head()
```

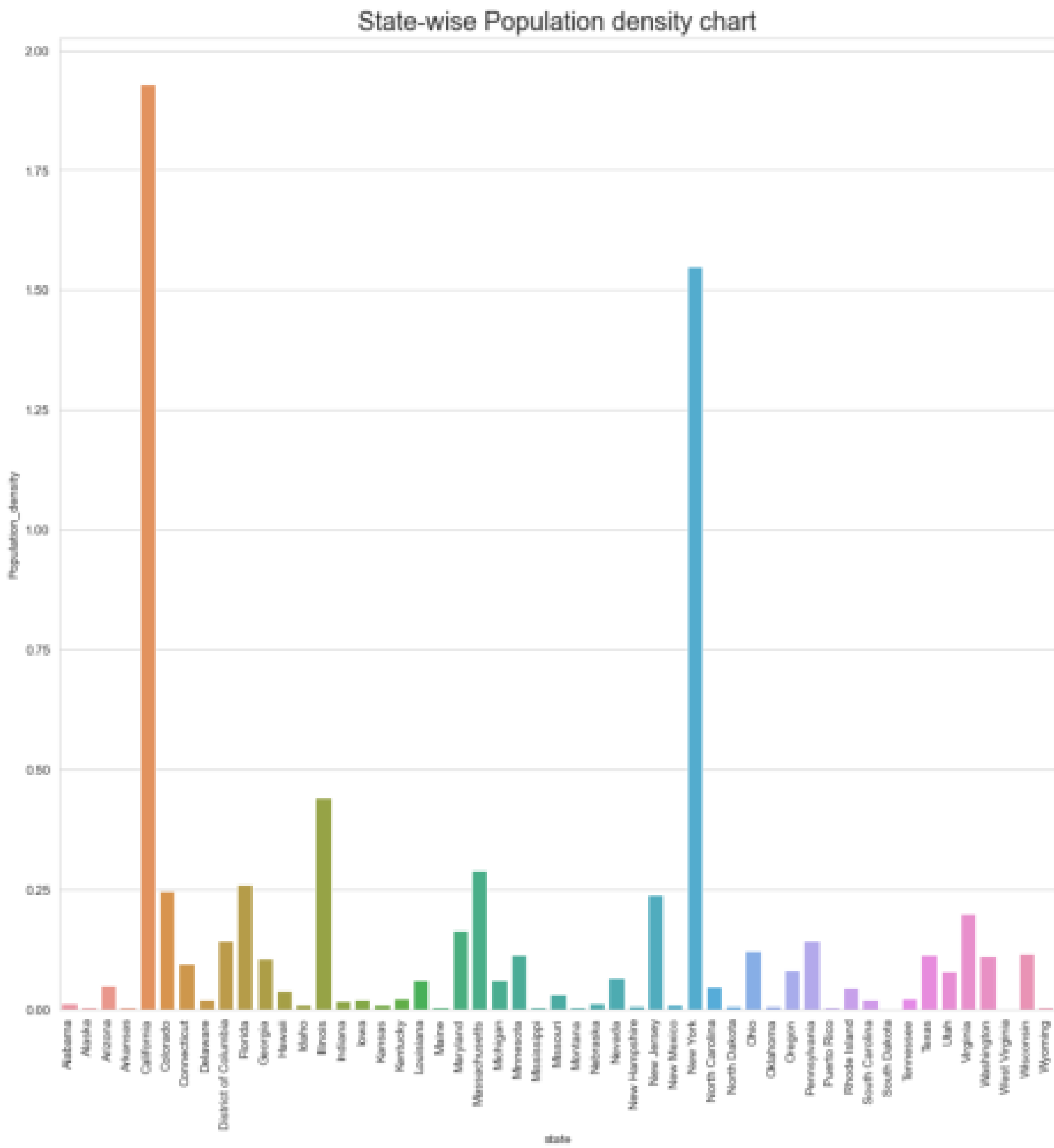
Out[39]:

	UID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	
14014	264403	140	31	34	New Jersey	NJ	Passaic	Garfield City	
3285	289712	140	147	51	Virginia	VA	Farmville	Farmville	
21706	222830	140	13	4	Arizona	AZ	Scottsdale	Tempe City	
11980	251185	140	27	25	Massachusetts	MA	Worcester	Worcester City	
12896	278178	140	101	42	Pennsylvania	PA	Philadelphia	Millbourne	Bo

5 rows × 83 columns

In [40]:
```
pop_density_gb = df_train1.groupby('state')['Population_density'].sum().reset_index()
```

In [41]:
```
plt.figure(figsize = (14,14))
sns.barplot(x = 'state', y = 'Population_density',data = pop_density_gb,orient='v').se
plt.title('State-wise Population density chart',fontsize=20)
plt.show()
```



The barplot shows the citywise population density

California and New York are more densely populated than other cities where as South Dakota is least densly populated

In [42]:
```
df_train1['median_age'] = (df_train1['male_age_median']*df_train1['male_pop'])+(df_tra
```

In [43]:
```
df_train1.head()
```

Out[43]:

	UID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	
14014	264403	140	31	34	New Jersey	NJ	Passaic	Garfield City	
3285	289712	140	147	51	Virginia	VA	Farmville	Farmville	
21706	222830	140	13	4	Arizona	AZ	Scottsdale	Tempe City	
11980	251185	140	27	25	Massachusetts	MA	Worcester	Worcester City	
12896	278178	140	101	42	Pennsylvania	PA	Philadelphia	Millbourne	Bo

5 rows × 84 columns

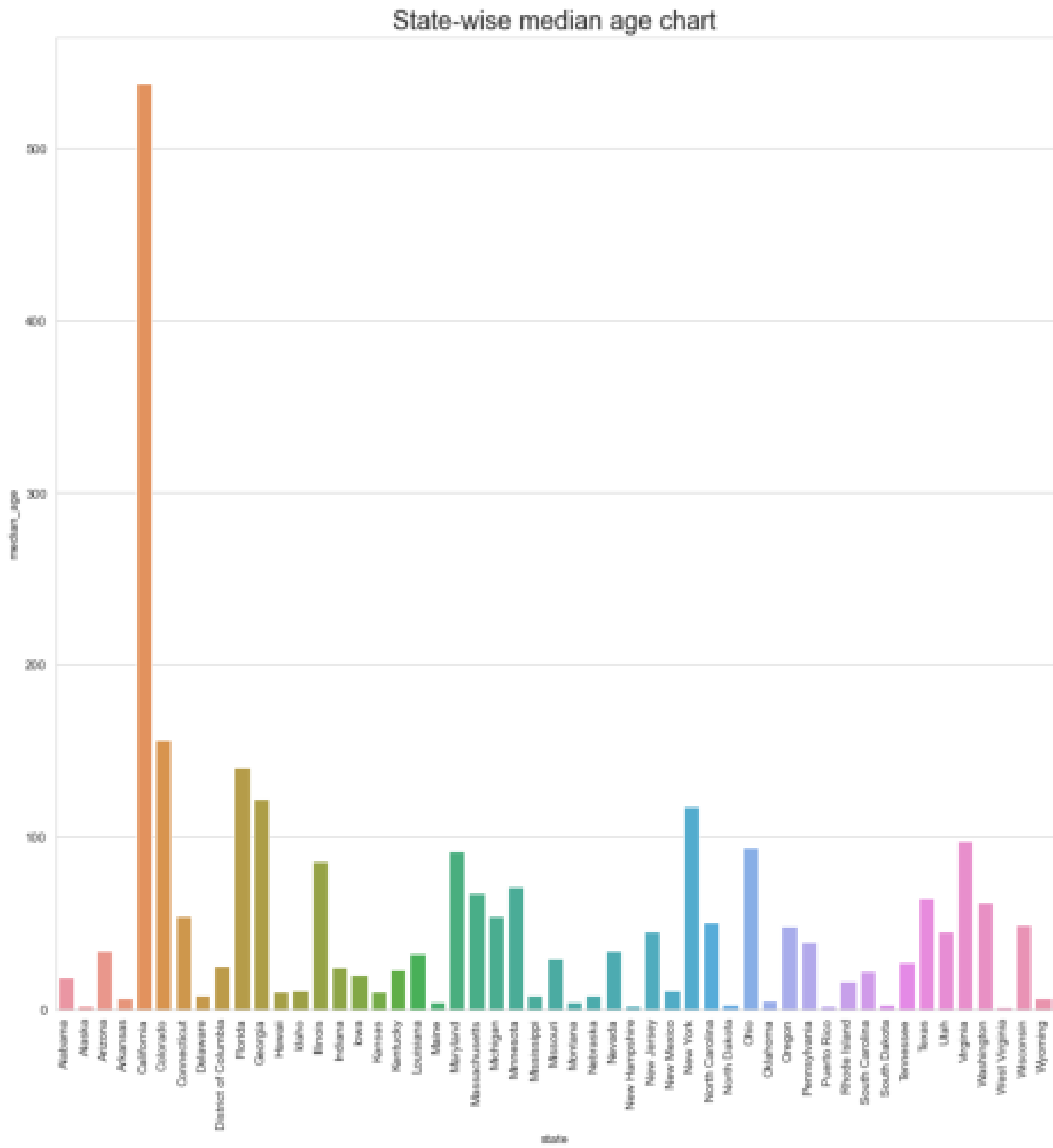
In [44]:
```
df_med_age = df_train1.groupby('state')['median_age'].size().reset_index()
```

In [45]:
```
df_med_age.head()
```

Out[45]:

	state	median_age
0	Alabama	18
1	Alaska	2
2	Arizona	34
3	Arkansas	6
4	California	538

In [46]:
```
plt.figure(figsize = (14,14))
sns.barplot(x='state',y='median_age',data=df_med_age).set_xticklabels(df_med_age['stat
plt.title('State-wise median age chart',fontsize=20)
plt.show()
```



California has highest median age as compared to other cities which means there are more elderly people living in california than other cities

In [47]:
```
df_train1.columns
```

Out[47]:
```
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', 'Bad_debt', 'Good_debt', 'remaining_income',
       'Population_density', 'median_age'],
      dtype='object')
```

In [48]:
```
df_for_age_analysis = df_train1[['state','city','place','pop','male_pop','female_pop',
```

In [49]:
```
df_for_age_analysis['male_age_median'].unique()
```

Out[49]:
```
array([30.5    , 19.25    , 29.91667, 30.75    , 21.25    , 20.66667,
       24.66667, 43.66667, 29.58333, 34.83333, 27.41667, 34.     ,
       29.08333, 39.33333, 32.     , 26.08333, 28.58333, 32.66667,
       28.5    , 31.5    , 23.16667, 36.5    , 33.66667, 65.33333,
       27.     , 39.25    , 41.16667, 31.08333, 39.66667, 36.     ,
       19.66667, 31.     , 28.     , 34.91667, 31.75    , 29.16667,
       28.41667, 39.91667, 30.16667, 35.33333, 19.33333, 32.25    ,
       36.66667, 39.41667, 25.08333, 34.66667, 21.83333, 30.91667,
       33.33333, 29.75    , 37.16667, 39.     , 28.75    , 24.25    ,
       32.83333, 24.5    , 25.66667, 30.33333, 43.41667, 32.08333,
       28.66667, 32.33333, 36.58333, 32.41667, 44.83333, 38.75    ,
       37.75    , 37.41667, 38.25    , 28.16667, 33.5    , 43.91667,
       40.41667, 33.25    , 30.41667, 29.5    , 49.66667, 34.5    ,
       35.25    , 31.33333, 41.83333, 30.     , 27.66667, 26.91667,
       40.33333, 34.58333, 25.91667, 35.41667, 30.08333, 26.58333,
       38.83333, 33.08333, 41.08333, 43.25    , 33.75    , 32.91667,
       35.08333, 27.25    , 24.     , 27.58333, 27.75    , 36.33333,
       35.91667, 28.33333, 45.16667, 30.25    , 35.5    , 38.66667,
       28.83333, 35.83333, 21.58333, 33.91667, 31.66667, 45.     ,
       31.58333, 27.91667, 42.25    , 42.16667, 27.16667, 20.83333,
       30.58333, 42.58333, 27.08333, 34.08333, 42.     , 30.83333,
       32.16667, 25.16667, 40.     , 26.33333, 40.58333, 29.66667,
       35.16667, 21.91667, 37.58333, 37.     , 33.58333, 23.25    ,
       38.16667, 35.58333, 18.75    , 22.5    , 32.5    , 39.58333,
       34.25    , 27.83333, 34.33333, 29.83333, 32.75    , 37.91667,
       34.41667, 24.58333, 35.     , 34.16667, 29.33333, 31.91667,
       50.33333, 36.25    , 49.91667, 36.75    , 24.41667, 38.91667,
       24.33333, 37.66667, 37.25    , 45.41667, 33.41667, 42.08333,
       28.08333, 33.83333, 44.     , 29.25    , 44.66667, 46.58333,
       37.83333, 31.16667, 36.91667, 24.91667, 38.58333, 25.83333,
       27.33333, 22.41667, 26.83333, 39.83333, 23.66667, 48.5    ,
       39.75    , 42.33333, 46.66667, 26.5    , 31.83333, 55.75    ,
       36.16667, 29.41667, 48.41667, 43.5    , 50.58333, 42.83333,
       21.66667, 38.08333, 37.5    , 42.91667, 39.16667, 41.75    ,
       34.75    , 44.16667, 38.5    , 35.75    , 24.75    , 32.58333,
       31.41667, 35.66667, 40.5    , 22.91667, 37.33333, 26.16667,
       33.16667, 47.25    , 22.25    , 26.     , 40.08333, 25.33333,
       29.     , 41.66667, 39.08333, 44.58333, 33.     , 26.25    ,
       42.5    , 45.83333, 26.66667, 28.25    , 36.41667, 40.75    ,
       25.41667, 41.5    , 38.     , 44.41667, 51.08333, 31.25    ,
       36.08333, 39.5    , 20.08333, 20.25    , 20.5    , 49.41667,
       25.     , 21.75    , 36.83333, 43.16667, 22.     , 27.5    ,
       24.83333, 40.91667, 41.     , 23.5    , 40.16667, 38.33333,
       45.75    , 60.91667, 24.08333, 43.08333, 50.41667, 42.75    ,
       41.41667, 41.58333, 42.41667, 30.66667, 45.91667, 51.5    ,
       15.08333, 47.16667, 47.41667, 49.83333, 23.33333, 21.33333,
       43.58333, 48.     , 50.66667, 44.5    , 23.58333, 40.83333,
       40.25    , 43.83333, 47.83333, 49.5    , 23.83333, 48.33333,
       47.33333, 37.08333, 43.33333, 44.75    , 45.5    , 41.25    ,
       24.16667, 15.91667, 49.25    , 48.83333, 22.33333, 41.33333,
       46.     , 23.     , 21.5    , 44.25    , 21.     , 45.66667,
       51.     , 43.75    , 44.08333, 38.41667, 28.91667, 18.91667,
       20.41667, 47.08333, 46.91667, 47.91667, 41.91667, 47.     ,
       51.58333, 49.08333, 46.08333, 44.33333, 25.25    , 26.75    ,
       59.25    , 16.16667, 60.66667, 45.25    , 60.08333, 52.16667,
       53.58333, 42.66667, 25.5    , 46.25    , 26.41667, 19.83333,
       22.83333, 49.16667, 25.58333, 57.25    , 44.91667, 64.83333,
       48.25    , 25.75    , 48.58333, 50.75    , 58.16667, 43.     ,
       52.25    , 46.5    , 40.66667, 45.33333, 51.33333, 54.25    ,
       48.66667, 15.16667, 20.16667, 48.75    , 58.08333, 21.08333,
       46.41667, 47.75    , 54.83333, 53.5    , 52.5    , 53.     ,
       46.83333, 47.66667, 58.91667, 46.16667, 21.16667, 48.08333,
       50.91667, 59.16667, 49.     , 46.33333, 17.41667])
```

In [50]:
```
df_for_age_analysis['male_pop_labels'] = pd.cut(df_for_age_analysis['male_age_median']
```

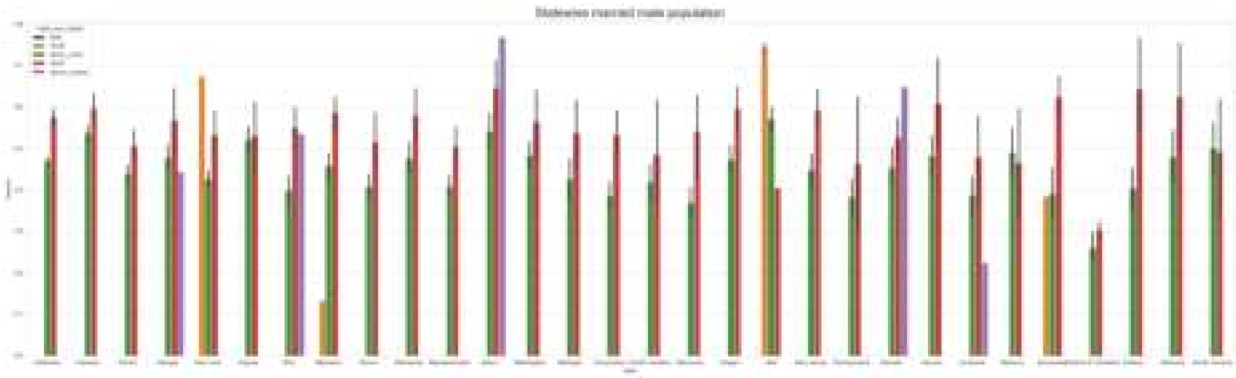
In [51]:
```
df_for_age_analysis['female_pop_labels'] = pd.cut(df_for_age_analysis['female_age_medi
```

In [52]:
```
df_for_age_analysis['state'].value_counts()[:30].index
```

Out[52]:
```
Index(['California', 'Colorado', 'Florida', 'Georgia', 'New York', 'Virginia',
       'Ohio', 'Maryland', 'Illinois', 'Minnesota', 'Massachusetts', 'Texas',
       'Washington', 'Michigan', 'Connecticut', 'North Carolina', 'Wisconsin',
       'Oregon', 'Utah', 'New Jersey', 'Pennsylvania', 'Nevada', 'Arizona',
       'Louisiana', 'Missouri', 'Tennessee', 'District of Columbia', 'Indiana',
       'Kentucky', 'South Carolina'],
      dtype='object')
```

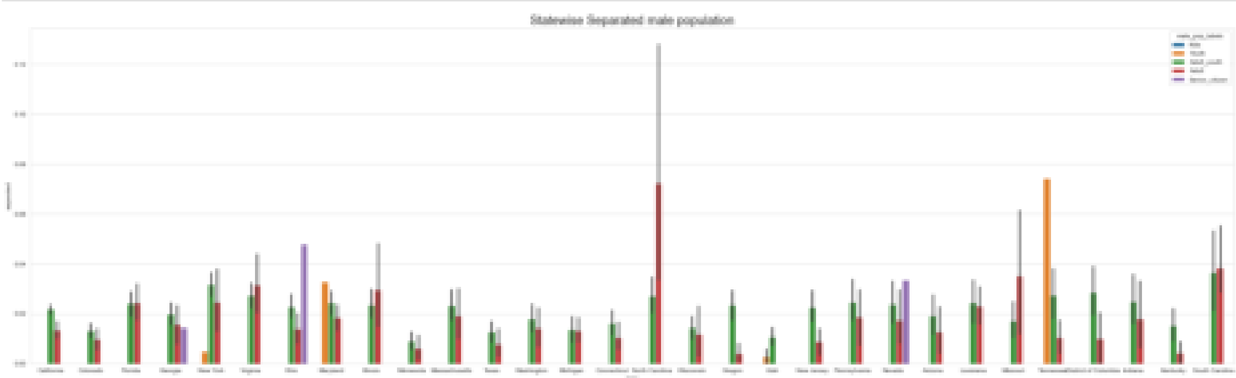
In [53]:

```
plt.figure(figsize=(35,10))
sns.barplot(x='state',y='married',data=df_for_age_analysis,hue='male_pop_labels',order
       'Ohio', 'Maryland', 'Illinois', 'Minnesota', 'Massachusetts', 'Texas',
       'Washington', 'Michigan', 'Connecticut', 'North Carolina', 'Wisconsin',
       'Oregon', 'Utah', 'New Jersey', 'Pennsylvania', 'Nevada', 'Arizona',
       'Louisiana', 'Missouri', 'Tennessee', 'District of Columbia', 'Indiana',
       'Kentucky', 'South Carolina'])
plt.title('Statewise married male population',fontsize=20)
plt.show()
```



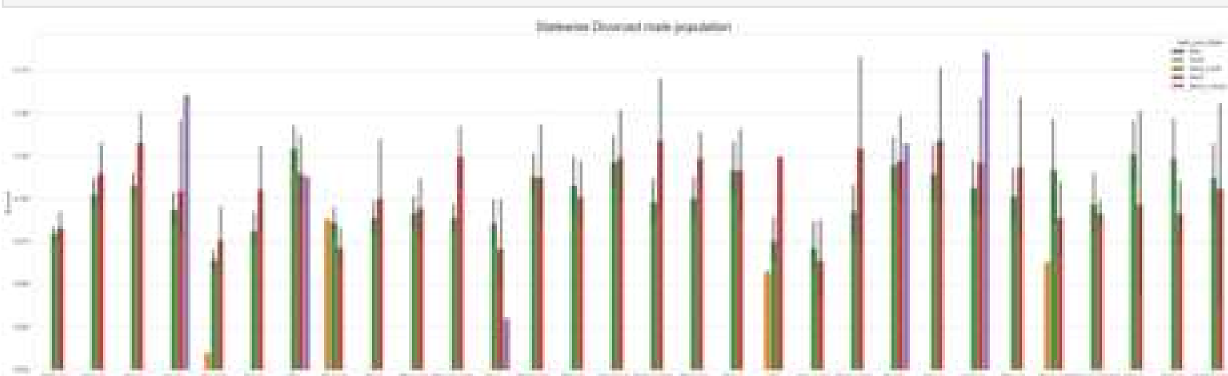
In [54]:

```
plt.figure(figsize=(35,10))
sns.barplot(x='state',y='separated',data=df_for_age_analysis,hue='male_pop_labels',ord
       'Ohio', 'Maryland', 'Illinois', 'Minnesota', 'Massachusetts', 'Texas',
       'Washington', 'Michigan', 'Connecticut', 'North Carolina', 'Wisconsin',
       'Oregon', 'Utah', 'New Jersey', 'Pennsylvania', 'Nevada', 'Arizona',
       'Louisiana', 'Missouri', 'Tennessee', 'District of Columbia', 'Indiana',
       'Kentucky', 'South Carolina'])
plt.title('Statewise Separated male population',fontsize=20)
plt.show()
```



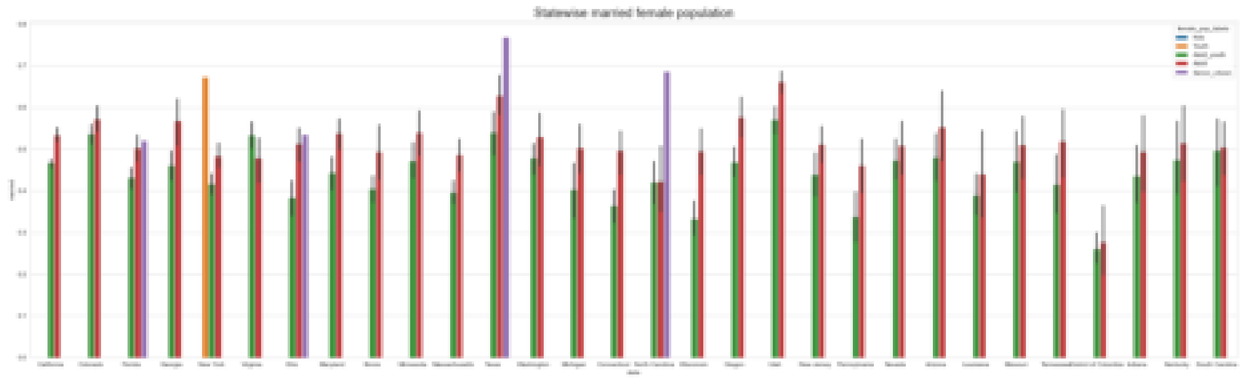
In [55]:

```
plt.figure(figsize=(35,10))
sns.barplot(x='state',y='divorced',data=df_for_age_analysis,hue='male_pop_labels',orde
       'Ohio', 'Maryland', 'Illinois', 'Minnesota', 'Massachusetts', 'Texas',
       'Washington', 'Michigan', 'Connecticut', 'North Carolina', 'Wisconsin',
       'Oregon', 'Utah', 'New Jersey', 'Pennsylvania', 'Nevada', 'Arizona',
       'Louisiana', 'Missouri', 'Tennessee', 'District of Columbia', 'Indiana',
       'Kentucky', 'South Carolina'])
plt.title('Statewise Divorced male population',fontsize=20)
plt.show()
```

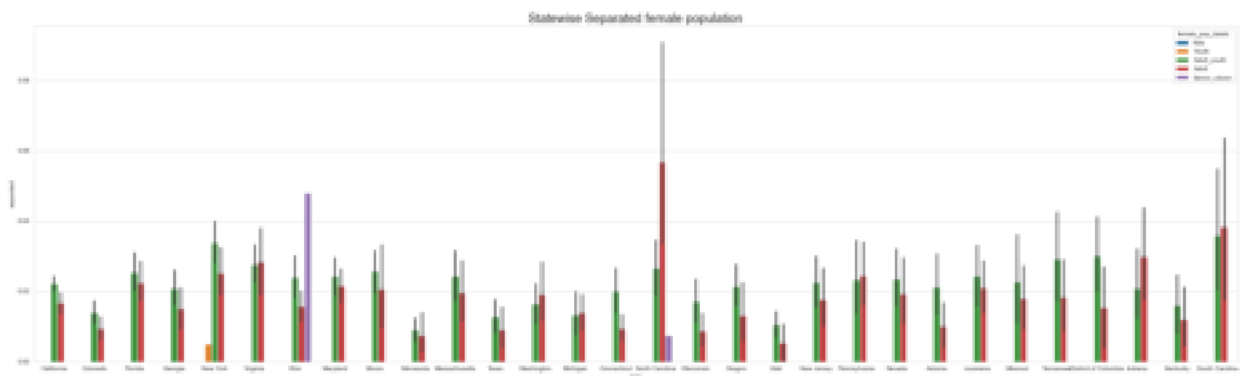


In [56]:

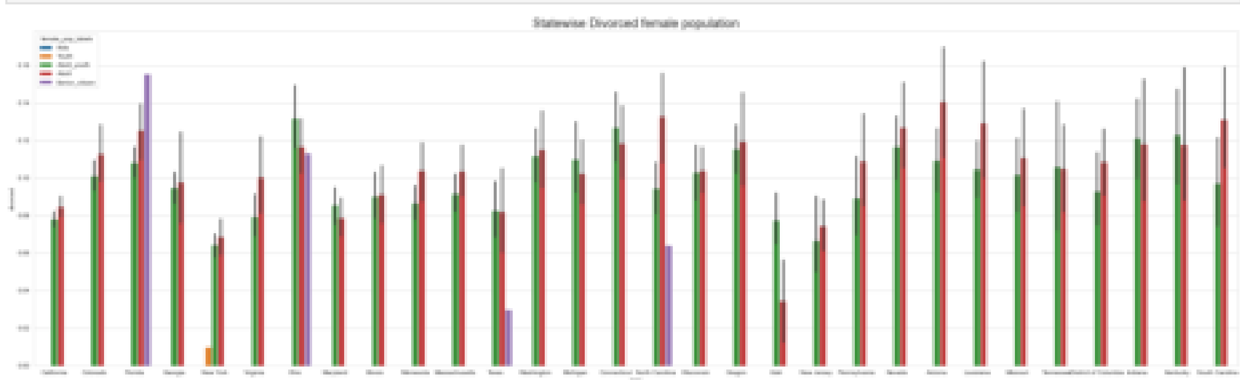
```
plt.figure(figsize=(35,10))
sns.barplot(x='state',y='married',data=df_for_age_analysis,hue='female_pop_labels',ord
       'Ohio', 'Maryland', 'Illinois', 'Minnesota', 'Massachusetts', 'Texas',
       'Washington', 'Michigan', 'Connecticut', 'North Carolina', 'Wisconsin',
       'Oregon', 'Utah', 'New Jersey', 'Pennsylvania', 'Nevada', 'Arizona',
       'Louisiana', 'Missouri', 'Tennessee', 'District of Columbia', 'Indiana',
       'Kentucky', 'South Carolina'])
plt.title('Statewise married female population',fontsize=20)
plt.show()
```



In [57]:
```
plt.figure(figsize=(35,10))
sns.barplot(x='state',y='separated',data=df_for_age_analysis,hue='female_pop_labels',c
       'Ohio', 'Maryland', 'Illinois', 'Minnesota', 'Massachusetts', 'Texas',
       'Washington', 'Michigan', 'Connecticut', 'North Carolina', 'Wisconsin',
       'Oregon', 'Utah', 'New Jersey', 'Pennsylvania', 'Nevada', 'Arizona',
       'Louisiana', 'Missouri', 'Tennessee', 'District of Columbia', 'Indiana',
       'Kentucky', 'South Carolina'])
plt.title('Statewise Separated female population',fontsize=20)
plt.show()
```



In [58]:
```
plt.figure(figsize=(35,10))
sns.barplot(x='state',y='divorced',data=df_for_age_analysis,hue='female_pop_labels',or
       'Ohio', 'Maryland', 'Illinois', 'Minnesota', 'Massachusetts', 'Texas',
       'Washington', 'Michigan', 'Connecticut', 'North Carolina', 'Wisconsin',
       'Oregon', 'Utah', 'New Jersey', 'Pennsylvania', 'Nevada', 'Arizona',
       'Louisiana', 'Missouri', 'Tennessee', 'District of Columbia', 'Indiana',
       'Kentucky', 'South Carolina'])
plt.title('Statewise Divorced female population',fontsize=20)
plt.show()
```



In [59]:
```
round(df_train1['rent_median'].sum()/df_train1['hi_median'].sum()*100,2)
```

Out[59]:
```
1.89
```

In [60]:
```
df_train1['rent%'] = round(df_train1['rent_median']/df_train1['hi_median']*100,2)
```

In [61]:
```
df_train1.head()
```

Out[61]:

	UID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	
14014	264403	140	31	34	New Jersey	NJ	Passaic	Garfield City	
3285	289712	140	147	51	Virginia	VA	Farmville	Farmville	
21706	222830	140	13	4	Arizona	AZ	Scottsdale	Tempe City	
11980	251185	140	27	25	Massachusetts	MA	Worcester	Worcester City	
12896	278178	140	101	42	Pennsylvania	PA	Philadelphia	Millbourne	Bo

5 rows × 85 columns

In [62]:

```
rent_df = df_train1.groupby('state')['rent%'].median().reset_index()
```

In [63]:

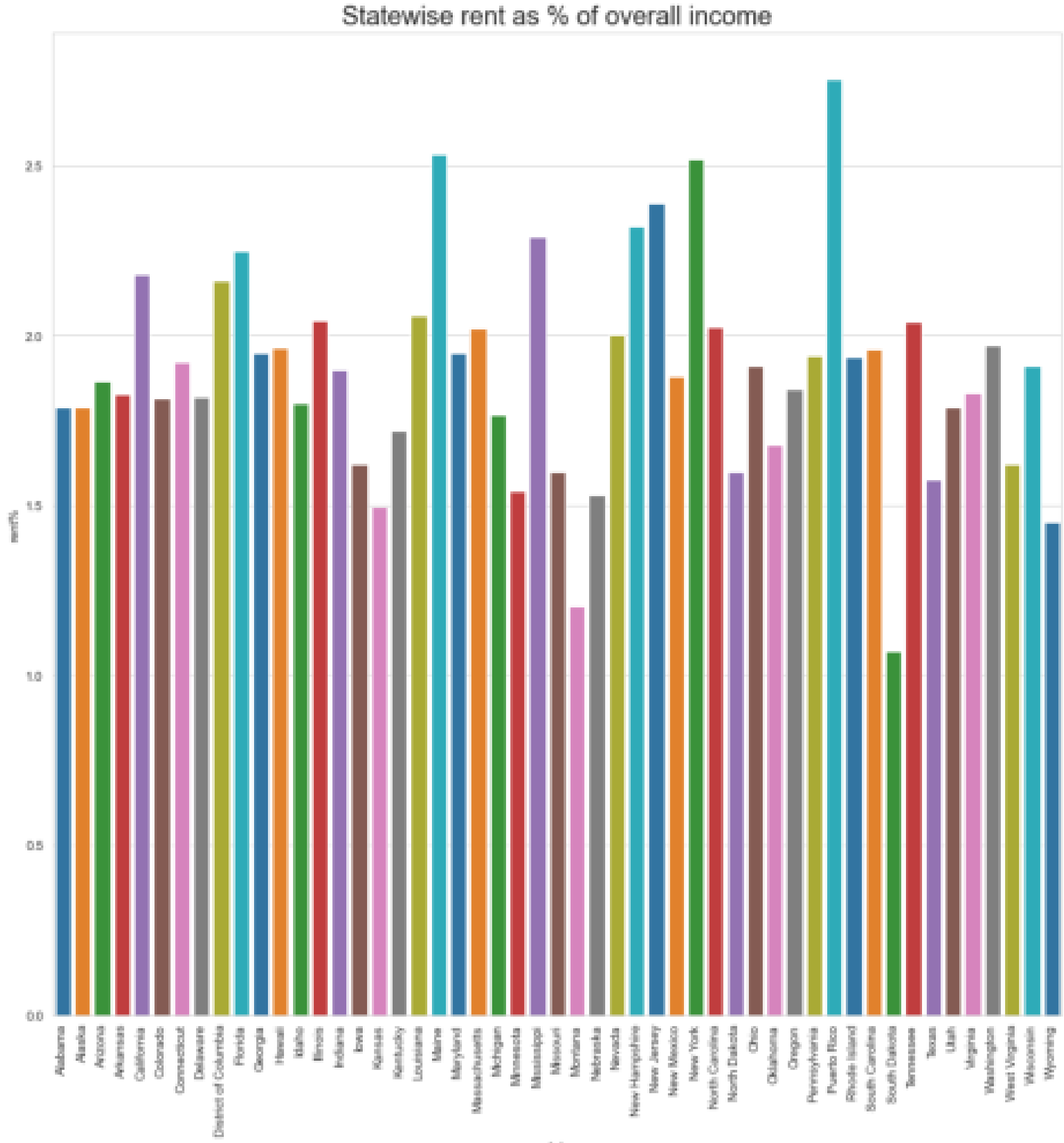
```
rent_df.head()
```

Out[63]:

	state	rent%
0	Alabama	1.790
1	Alaska	1.790
2	Arizona	1.865
3	Arkansas	1.825
4	California	2.180

In [64]:

```
plt.figure(figsize=(14,14))
sns.barplot(x='state',y='rent%',data=rent_df,palette='tab10').set_xticklabels(rent_df[
plt.title('Statewise rent as % of overall income',fontsize=20)
plt.show()
```

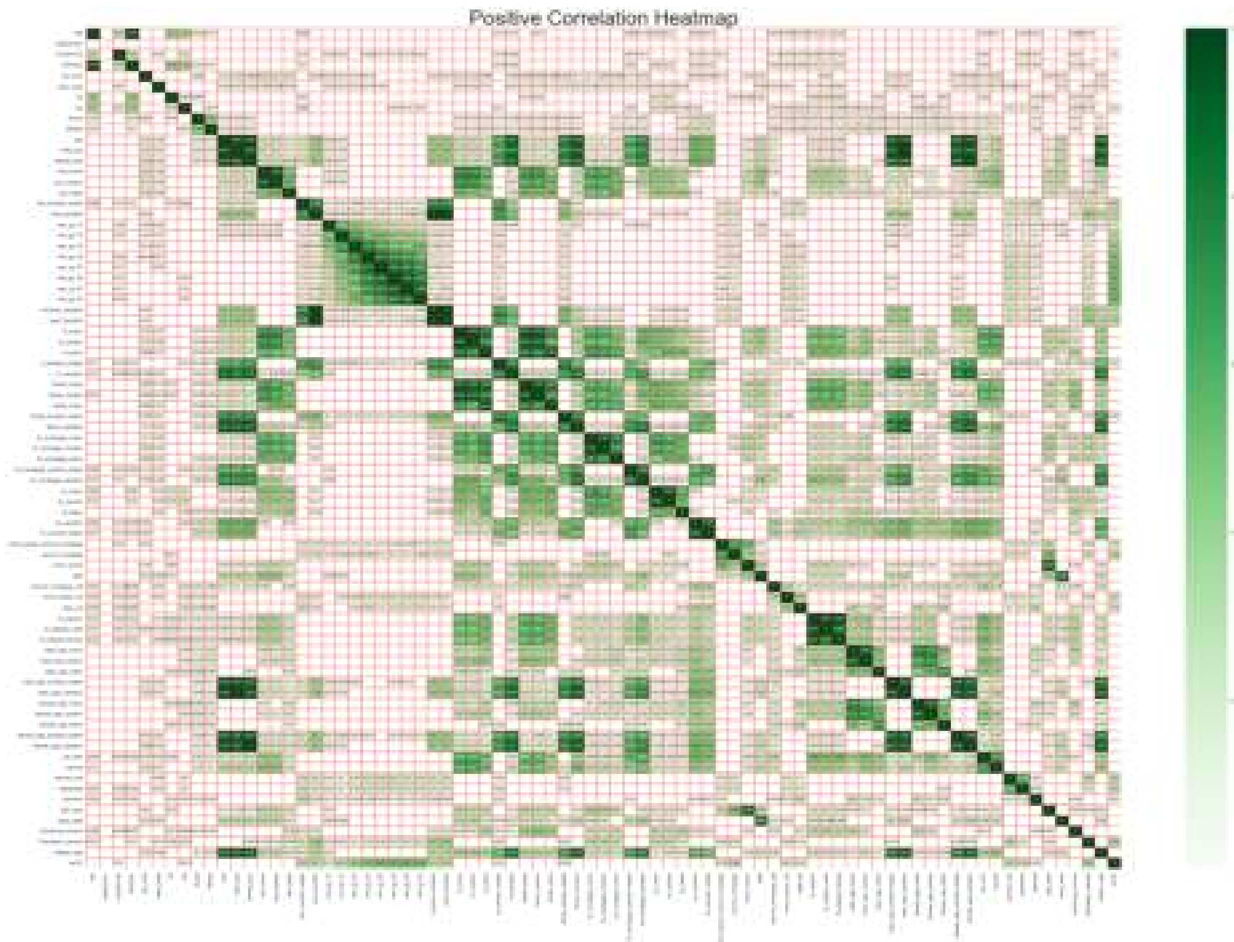


People from Puerto Rico are having less income and paying most rent as percentage of their income where as South Dakota people are having less rent% as their income

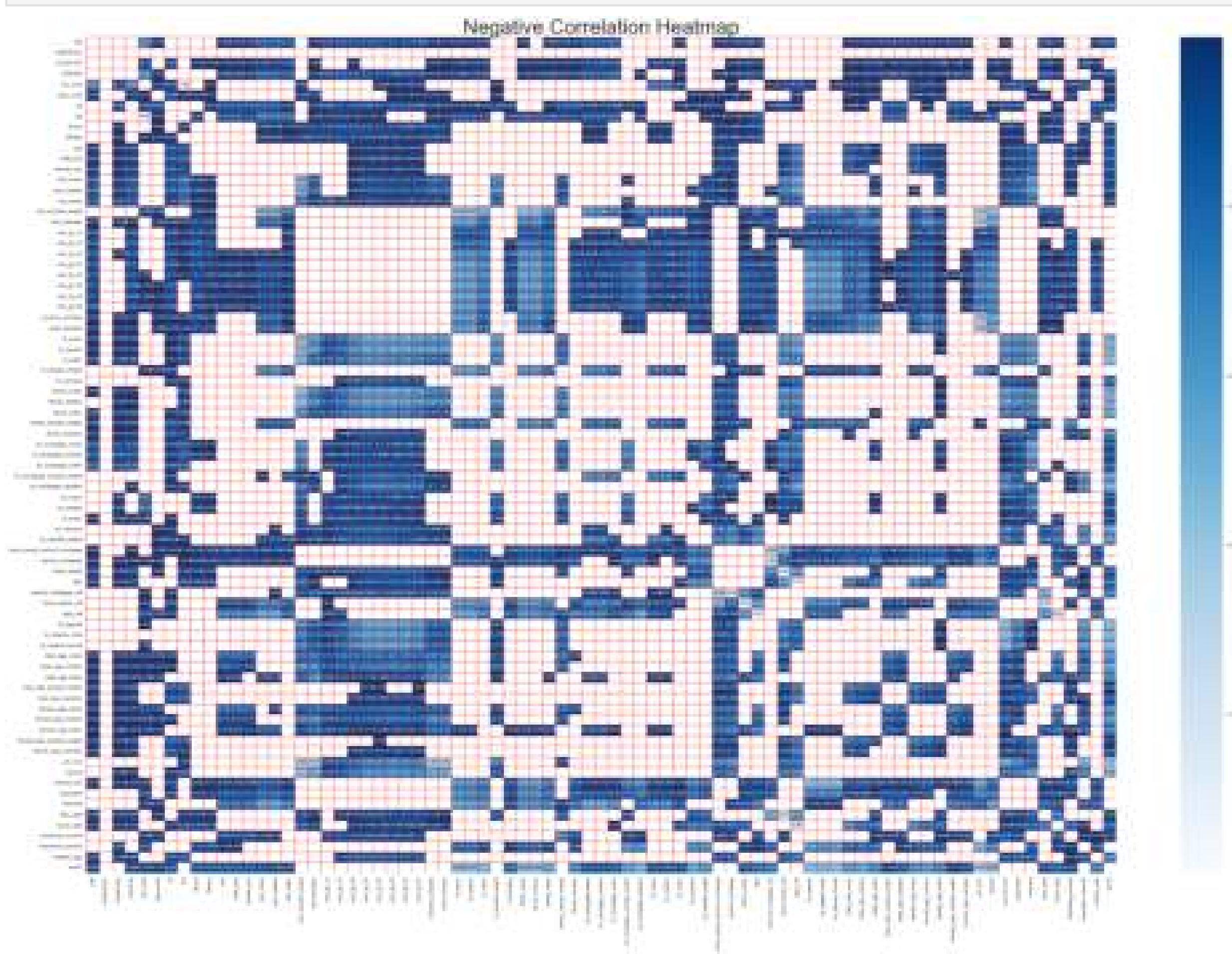
In [65]:
```
corr = df_train1.corr()
```

In [66]:
```
positive_correlation = corr[corr>=0]
negative_correlation = corr[corr<0]
```

In [67]:
```
plt.figure(figsize = (45,30))
sns.heatmap(positive_correlation,cmap='Greens',annot=True,linecolor='red',linewidths=1
plt.title('Positive Correlation Heatmap',fontsize=40)
plt.show()
```



In [68]:
```
plt.figure(figsize = (45,30))
sns.heatmap(negative_correlation,cmap='Blues',annot=True,linecolor='red',linewidths=1)
plt.title('Negative Correlation Heatmap',fontsize=40)
plt.show()
```



Data Preprocessing

In [69]:
```
df_train1.describe()
```

Out[69]:

	UID	SUMLEVEL	COUNTYID	STATEID	zip_code	area_code	lat
count	2500.000000	2500.0	2500.000000	2500.000000	2500.000000	2500.000000	2500.000000
mean	251192.994400	140.0	75.166400	23.252400	55915.810400	602.945600	37.862908
std	21841.694936	0.0	100.094679	16.302485	31729.029582	228.898513	4.785496
min	220366.000000	140.0	1.000000	1.000000	951.000000	201.000000	18.384790
25%	232083.250000	140.0	31.000000	8.000000	29280.500000	407.000000	34.055419
50%	246485.000000	140.0	53.000000	19.000000	55337.000000	626.000000	38.831048
75%	269242.250000	140.0	89.000000	37.000000	90056.000000	775.000000	41.129508
max	294317.000000	140.0	820.000000	72.000000	99701.000000	989.000000	64.851287

8 rows × 79 columns

In [70]:

```
df_train1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2500 entries, 14014 to 22594
Data columns (total 85 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   UID                          2500 non-null   int64
 1   SUMLEVEL                     2500 non-null   int64
 2   COUNTYID                     2500 non-null   int64
 3   STATEID                      2500 non-null   int64
 4   state                        2500 non-null   object
 5   state_ab                     2500 non-null   object
 6   city                         2500 non-null   object
 7   place                        2500 non-null   object
 8   type                         2500 non-null   object
 9   primary                      2500 non-null   object
 10  zip_code                     2500 non-null   int64
 11  area_code                    2500 non-null   int64
 12  lat                          2500 non-null   float64
 13  lng                          2500 non-null   float64
 14  ALand                        2500 non-null   float64
 15  AWater                       2500 non-null   int64
 16  pop                          2500 non-null   int64
 17  male_pop                     2500 non-null   int64
 18  female_pop                   2500 non-null   int64
 19  rent_mean                    2500 non-null   float64
 20  rent_median                  2500 non-null   float64
 21  rent_stdev                   2500 non-null   float64
 22  rent_sample_weight           2500 non-null   float64
 23  rent_samples                 2500 non-null   float64
 24  rent_gt_10                   2500 non-null   float64
 25  rent_gt_15                   2500 non-null   float64
 26  rent_gt_20                   2500 non-null   float64
 27  rent_gt_25                   2500 non-null   float64
 28  rent_gt_30                   2500 non-null   float64
 29  rent_gt_35                   2500 non-null   float64
 30  rent_gt_40                   2500 non-null   float64
 31  rent_gt_50                   2500 non-null   float64
 32  universe_samples             2500 non-null   int64
 33  used_samples                 2500 non-null   int64
 34  hi_mean                      2500 non-null   float64
 35  hi_median                    2500 non-null   float64
 36  hi_stdev                     2500 non-null   float64
 37  hi_sample_weight             2500 non-null   float64
 38  hi_samples                   2500 non-null   float64
 39  family_mean                  2500 non-null   float64
 40  family_median                2500 non-null   float64
 41  family_stdev                 2500 non-null   float64
 42  family_sample_weight         2500 non-null   float64
 43  family_samples               2500 non-null   float64
 44  hc_mortgage_mean             2500 non-null   float64
 45  hc_mortgage_median           2500 non-null   float64
 46  hc_mortgage_stdev            2500 non-null   float64
 47  hc_mortgage_sample_weight    2500 non-null   float64
 48  hc_mortgage_samples          2500 non-null   float64
 49  hc_mean                      2500 non-null   float64
 50  hc_median                    2500 non-null   float64
 51  hc_stdev                     2500 non-null   float64
 52  hc_samples                   2500 non-null   float64
 53  hc_sample_weight             2500 non-null   float64
 54  home_equity_second_mortgage  2500 non-null   float64
 55  second_mortgage              2500 non-null   float64
 56  home_equity                  2500 non-null   float64
 57  debt                         2500 non-null   float64
 58  second_mortgage_cdf          2500 non-null   float64
 59  home_equity_cdf              2500 non-null   float64
 60  debt_cdf                     2500 non-null   float64
 61  hs_degree                    2500 non-null   float64
 62  hs_degree_male               2500 non-null   float64
 63  hs_degree_female             2500 non-null   float64
 64  male_age_mean                2500 non-null   float64
 65  male_age_median              2500 non-null   float64
 66  male_age_stdev               2500 non-null   float64
 67  male_age_sample_weight       2500 non-null   float64
 68  male_age_samples             2500 non-null   float64
 69  female_age_mean              2500 non-null   float64
 70  female_age_median            2500 non-null   float64
 71  female_age_stdev             2500 non-null   float64
 72  female_age_sample_weight     2500 non-null   float64
```

```
 73   female_age_samples           2500 non-null   float64
 74   pct_own                      2500 non-null   float64
 75   married                      2500 non-null   float64
 76   married_snp                  2500 non-null   float64
 77   separated                    2500 non-null   float64
 78   divorced                     2500 non-null   float64
 79   Bad_debt                     2500 non-null   float64
 80   Good_debt                    2500 non-null   float64
 81   remaining_income             2500 non-null   float64
 82   Population_density           2500 non-null   float64
 83   median_age                   2500 non-null   float64
 84   rent%                        2500 non-null   float64
dtypes: float64(67), int64(12), object(6)
memory usage: 1.6+ MB
```

In [71]:
```
numerical_variables = df_train1.select_dtypes(('int64','float64'))
```

In [72]:
```
numerical_variables.shape
```

Out[72]:
```
(2500, 79)
```

In [73]:
```
numerical_variables.drop(['SUMLEVEL','lat','lng','ALand','AWater'],axis=1,inplace=True
```

In [74]:
```
numerical_variables.shape
```

Out[74]:
```
(2500, 74)
```

In [75]:
```
from sklearn.decomposition import FactorAnalysis
fa = FactorAnalysis(n_components=25)
```

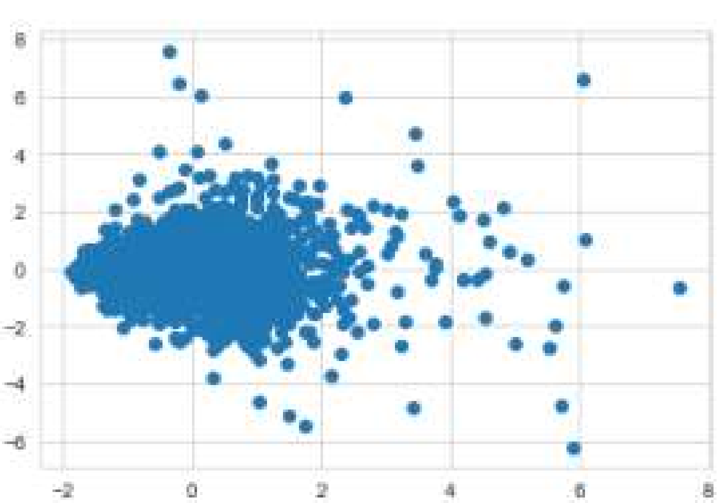
In [76]:
```
fact = fa.fit_transform(numerical_variables)
```

In [77]:
```
fact
```

Out[77]:
```
array([[ 0.18747801,  0.43370307, -1.30400813, ..., -1.70960839,
        -1.18026485,  1.08930948],
       [-1.1948433 , -1.44585335, -0.33787539, ..., -2.26756953,
        -2.93730713,  2.23589699],
       [-1.1257653 ,  0.50189866, -0.30828815, ..., -2.10901491,
        -1.61832654,  1.54754634],
       ...,
       [ 1.83548715,  1.27863704, -0.01405628, ..., -0.20123355,
        -0.09886889, -0.59632592],
       [ 2.41430476,  0.3349158 ,  0.1555483 , ...,  1.03349932,
         0.09597458,  0.34153176],
       [-0.5501776 , -0.39648252, -0.38473123, ..., -0.6043464 ,
        -1.09735072, -0.03229883]])
```

In [78]:
```
plt.scatter(fact[:,0],fact[:,1])
```

Out[78]:
```
<matplotlib.collections.PathCollection at 0x1b2638f0190>
```



In [79]:
```
variables = pd.DataFrame(fact)
```

In [80]:
```
variables.head()
```

Out[80]:

	0	1	2	3	4	5	6	7	8
0	0.187478	0.433703	-1.304008	-0.282134	1.636429	-0.180420	-0.221500	0.124807	-0.098687
1	-1.194843	-1.445853	-0.337875	2.386225	0.749854	1.702807	-2.781141	-0.635829	-1.123816
2	-1.125765	0.501899	-0.308288	0.807927	-1.287159	-0.783129	-0.702025	0.123263	-0.792216
3	-1.118196	0.465698	-0.839532	0.341571	1.481207	-0.928418	-0.173575	0.130064	0.117703
4	0.969968	0.398325	-1.235073	2.961507	1.217279	0.697137	-1.268541	2.825097	1.948362

5 rows × 25 columns

In [81]:

```
numerical_variables.isnull().sum()
```

Out[81]:

```
UID                   0
COUNTYID              0
STATEID               0
zip_code              0
area_code             0
                     ..
Good_debt             0
remaining_income      0
Population_density    0
median_age            0
rent%                 0
Length: 74, dtype: int64
```

In [82]:

```
numerical_variables['hc_mortgage_mean'].isnull().sum()
```

Out[82]:

```
0
```

In [83]:

```
numerical_variables.columns
```

Out[83]:

```
Index(['UID', 'COUNTYID', 'STATEID', 'zip_code', 'area_code', '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', 'remaining_income', 'Population_density',
       'median_age', 'rent%'],
      dtype='object')
```

In [84]:

```
x = numerical_variables[['UID', 'COUNTYID', 'STATEID', 'zip_code', 'area_code', '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_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', 'remaining_income', 'Population_density',
       'median_age', 'rent%']]
y = numerical_variables['hc_mortgage_mean']
```

Splitting the data as train and test

In [85]:
```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=0.3,random_state=50)
```

In [86]:
```
x_train.shape
```

Out[86]:
```
(1750, 73)
```

In [87]:
```
x_test.shape
```

Out[87]:
```
(750, 73)
```

In [88]:
```
y_train.shape
```

Out[88]:
```
(1750,)
```

In [89]:
```
y_test.shape
```

Out[89]:
```
(750,)
```

Here we are using multi-linear regression model

In [90]:
```
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
```

In [91]:
```
Model = lm.fit(x_train,y_train)
```

In [92]:
```
y_pred = lm.predict(x_test)
```

In [93]:
```
from sklearn.metrics import r2_score,mean_absolute_error,mean_squared_error
```

In [94]:
```
MAE = mean_absolute_error(y_test,y_pred)
MAE
```

Out[94]:
```
52.13954704130099
```

In [95]:
```
MSE = mean_squared_error(y_test,y_pred)
MSE
```

Out[95]:
```
5863.029336911928
```

In [96]:
```
RMSE = np.sqrt(MSE)
RMSE
```

Out[96]:
```
76.57042077011154
```

In [97]:
```
r2 = r2_score(y_test,y_pred)
r2
```

Out[97]:
```
0.9827880351606979
```

We got 98.27% r2 score which is above acceptance limit so we can skip the remaining steps now we have to predict the valuse for hc_mortgage_mean for the test dataset

In [98]:
```
df_test = pd.read_csv('T:\Masters In Data Science\Capstone Project\Project 1\\test.csv
```

In [99]:
```
df_test.head()
```

Out[99]:

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

5 rows × 80 columns

In [100…

```
df_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11709 entries, 0 to 11708
Data columns (total 80 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   UID                          11709 non-null  int64
 1   BLOCKID                      0 non-null      float64
 2   SUMLEVEL                     11709 non-null  int64
 3   COUNTYID                     11709 non-null  int64
 4   STATEID                      11709 non-null  int64
 5   state                        11709 non-null  object
 6   state_ab                     11709 non-null  object
 7   city                         11709 non-null  object
 8   place                        11709 non-null  object
 9   type                         11709 non-null  object
 10  primary                      11709 non-null  object
 11  zip_code                     11709 non-null  int64
 12  area_code                    11709 non-null  int64
 13  lat                          11709 non-null  float64
 14  lng                          11709 non-null  float64
 15  ALand                        11709 non-null  int64
 16  AWater                       11709 non-null  int64
 17  pop                          11709 non-null  int64
 18  male_pop                     11709 non-null  int64
 19  female_pop                   11709 non-null  int64
 20  rent_mean                    11561 non-null  float64
 21  rent_median                  11561 non-null  float64
 22  rent_stdev                   11561 non-null  float64
 23  rent_sample_weight           11561 non-null  float64
 24  rent_samples                 11561 non-null  float64
 25  rent_gt_10                   11560 non-null  float64
 26  rent_gt_15                   11560 non-null  float64
 27  rent_gt_20                   11560 non-null  float64
 28  rent_gt_25                   11560 non-null  float64
 29  rent_gt_30                   11560 non-null  float64
 30  rent_gt_35                   11560 non-null  float64
 31  rent_gt_40                   11560 non-null  float64
 32  rent_gt_50                   11560 non-null  float64
 33  universe_samples             11709 non-null  int64
 34  used_samples                 11709 non-null  int64
 35  hi_mean                      11587 non-null  float64
 36  hi_median                    11587 non-null  float64
 37  hi_stdev                     11587 non-null  float64
 38  hi_sample_weight             11587 non-null  float64
 39  hi_samples                   11587 non-null  float64
 40  family_mean                  11573 non-null  float64
 41  family_median                11573 non-null  float64
 42  family_stdev                 11573 non-null  float64
 43  family_sample_weight         11573 non-null  float64
 44  family_samples               11573 non-null  float64
 45  hc_mortgage_mean             11441 non-null  float64
 46  hc_mortgage_median           11441 non-null  float64
 47  hc_mortgage_stdev            11441 non-null  float64
 48  hc_mortgage_sample_weight    11441 non-null  float64
 49  hc_mortgage_samples          11441 non-null  float64
 50  hc_mean                      11419 non-null  float64
 51  hc_median                    11419 non-null  float64
 52  hc_stdev                     11419 non-null  float64
 53  hc_samples                   11419 non-null  float64
 54  hc_sample_weight             11419 non-null  float64
 55  home_equity_second_mortgage  11489 non-null  float64
 56  second_mortgage              11489 non-null  float64
 57  home_equity                  11489 non-null  float64
 58  debt                         11489 non-null  float64
 59  second_mortgage_cdf          11489 non-null  float64
 60  home_equity_cdf              11489 non-null  float64
 61  debt_cdf                     11489 non-null  float64
 62  hs_degree                    11624 non-null  float64
 63  hs_degree_male               11620 non-null  float64
 64  hs_degree_female             11604 non-null  float64
 65  male_age_mean                11625 non-null  float64
 66  male_age_median              11625 non-null  float64
 67  male_age_stdev               11625 non-null  float64
 68  male_age_sample_weight       11625 non-null  float64
 69  male_age_samples             11625 non-null  float64
 70  female_age_mean              11613 non-null  float64
 71  female_age_median            11613 non-null  float64
 72  female_age_stdev             11613 non-null  float64
```

```
 73   female_age_sample_weight     11613 non-null   float64
 74   female_age_samples           11613 non-null   float64
 75   pct_own                      11587 non-null   float64
 76   married                      11625 non-null   float64
 77   married_snp                  11625 non-null   float64
 78   separated                    11625 non-null   float64
 79   divorced                     11625 non-null   float64
dtypes: float64(61), int64(13), object(6)
memory usage: 7.1+ MB
```

In [101…
```
df_test.isnull().sum()
```

Out[101]:
```
UID                0
BLOCKID        11709
SUMLEVEL           0
COUNTYID           0
STATEID            0
               ...  
pct_own          122
married           84
married_snp       84
separated         84
divorced          84
Length: 80, dtype: int64
```

In [102…
```
df_test = df_test.drop(['BLOCKID'],axis=1)
```

In [103…
```
df_test.isnull().sum()
```

Out[103]:
```
UID              0
SUMLEVEL         0
COUNTYID         0
STATEID          0
state            0
              ... 
pct_own        122
married         84
married_snp     84
separated       84
divorced        84
Length: 79, dtype: int64
```

In [104…
```
df_test.dropna(inplace=True)
```

In [105…
```
df_test.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 11355 entries, 0 to 11708
Data columns (total 79 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   UID                          11355 non-null  int64  
 1   SUMLEVEL                     11355 non-null  int64  
 2   COUNTYID                     11355 non-null  int64  
 3   STATEID                      11355 non-null  int64  
 4   state                        11355 non-null  object 
 5   state_ab                     11355 non-null  object 
 6   city                         11355 non-null  object 
 7   place                        11355 non-null  object 
 8   type                         11355 non-null  object 
 9   primary                      11355 non-null  object 
 10  zip_code                     11355 non-null  int64  
 11  area_code                    11355 non-null  int64  
 12  lat                          11355 non-null  float64
 13  lng                          11355 non-null  float64
 14  ALand                        11355 non-null  int64  
 15  AWater                       11355 non-null  int64  
 16  pop                          11355 non-null  int64  
 17  male_pop                     11355 non-null  int64  
 18  female_pop                   11355 non-null  int64  
 19  rent_mean                    11355 non-null  float64
 20  rent_median                  11355 non-null  float64
 21  rent_stdev                   11355 non-null  float64
 22  rent_sample_weight           11355 non-null  float64
 23  rent_samples                 11355 non-null  float64
 24  rent_gt_10                   11355 non-null  float64
 25  rent_gt_15                   11355 non-null  float64
 26  rent_gt_20                   11355 non-null  float64
 27  rent_gt_25                   11355 non-null  float64
 28  rent_gt_30                   11355 non-null  float64
 29  rent_gt_35                   11355 non-null  float64
 30  rent_gt_40                   11355 non-null  float64
 31  rent_gt_50                   11355 non-null  float64
 32  universe_samples             11355 non-null  int64  
 33  used_samples                 11355 non-null  int64  
 34  hi_mean                      11355 non-null  float64
 35  hi_median                    11355 non-null  float64
 36  hi_stdev                     11355 non-null  float64
 37  hi_sample_weight             11355 non-null  float64
 38  hi_samples                   11355 non-null  float64
 39  family_mean                  11355 non-null  float64
 40  family_median                11355 non-null  float64
 41  family_stdev                 11355 non-null  float64
 42  family_sample_weight         11355 non-null  float64
 43  family_samples               11355 non-null  float64
 44  hc_mortgage_mean             11355 non-null  float64
 45  hc_mortgage_median           11355 non-null  float64
 46  hc_mortgage_stdev            11355 non-null  float64
 47  hc_mortgage_sample_weight    11355 non-null  float64
 48  hc_mortgage_samples          11355 non-null  float64
 49  hc_mean                      11355 non-null  float64
 50  hc_median                    11355 non-null  float64
 51  hc_stdev                     11355 non-null  float64
 52  hc_samples                   11355 non-null  float64
 53  hc_sample_weight             11355 non-null  float64
 54  home_equity_second_mortgage  11355 non-null  float64
 55  second_mortgage              11355 non-null  float64
 56  home_equity                  11355 non-null  float64
 57  debt                         11355 non-null  float64
 58  second_mortgage_cdf          11355 non-null  float64
 59  home_equity_cdf              11355 non-null  float64
 60  debt_cdf                     11355 non-null  float64
 61  hs_degree                    11355 non-null  float64
 62  hs_degree_male               11355 non-null  float64
 63  hs_degree_female             11355 non-null  float64
 64  male_age_mean                11355 non-null  float64
 65  male_age_median              11355 non-null  float64
 66  male_age_stdev               11355 non-null  float64
 67  male_age_sample_weight       11355 non-null  float64
 68  male_age_samples             11355 non-null  float64
 69  female_age_mean              11355 non-null  float64
 70  female_age_median            11355 non-null  float64
 71  female_age_stdev             11355 non-null  float64
 72  female_age_sample_weight     11355 non-null  float64
```

```
 73   female_age_samples          11355 non-null  float64
 74   pct_own                     11355 non-null  float64
 75   married                     11355 non-null  float64
 76   married_snp                 11355 non-null  float64
 77   separated                   11355 non-null  float64
 78   divorced                    11355 non-null  float64
dtypes: float64(60), int64(13), object(6)
memory usage: 6.9+ MB
```

In [106…
```
df_test1 = df_test.nlargest(2500,['second_mortgage','pct_own'])
```

In [107…
```
df_test1.shape
```

Out[107]:
```
(2500, 79)
```

In [108…
```
df_test1['Bad_debt'] = df_test1['second_mortgage'] + df_test1['home_equity'] - df_test
df_test1['Good_debt'] = df_test1['debt'] - df_test1['Bad_debt']
```

In [109…
```
df_test1.describe()
```

Out[109]:

	UID	SUMLEVEL	COUNTYID	STATEID	zip_code	area_code	lat
count	2500.000000	2500.0	2500.000000	2500.000000	2500.000000	2500.000000	2500.000000
mean	253296.506800	140.0	79.058000	24.922000	55553.227200	593.673200	37.914653
std	22355.084714	0.0	104.577449	16.760247	31594.881314	228.389981	4.900106
min	220352.000000	140.0	1.000000	1.000000	725.000000	201.000000	18.226608
25%	232274.000000	140.0	31.000000	8.000000	28030.000000	405.000000	34.081025
50%	251218.000000	140.0	57.000000	24.000000	55104.000000	615.000000	38.787609
75%	272187.250000	140.0	95.000000	39.000000	90029.000000	773.000000	41.298734
max	294285.000000	140.0	810.000000	72.000000	99705.000000	989.000000	64.758471

8 rows × 75 columns

In [110…
```
df_test1['Remaining_income'] = df_test1['family_median'] - df_test1['hi_median']
```

In [111…
```
df_test1['Population_density'] = df_test1['pop'] / df_test1['ALand']
```

In [112…
```
df_test1['median_age'] = (df_test1['male_age_median']*df_test1['male_pop'])+(df_test1[
```

In [113…
```
df_test1['rent%'] = round(df_test1['rent_median']/df_test1['hi_median']*100,2)
```

In [114…
```
df_test1.head()
```

Out[114]:

	UID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	
6238	266140	140	5	36	New York	NY	Bronx	Mount Vernon City	
9088	248877	140	33	22	Louisiana	LA	Baton Rouge	Port Allen City	
8771	254689	140	125	26	Michigan	MI	Southfield	Oak Park City	
4976	252317	140	33	24	Maryland	MD	Adelphi	Adelphi	
11051	278176	140	101	42	Pennsylvania	PA	Philadelphia	Millbourne	Borc

5 rows × 85 columns

In [115…
```
df_test1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2500 entries, 6238 to 6099
Data columns (total 85 columns):
 #   Column                       Non-Null Count  Dtype  
---  ------                       --------------  -----  
 0   UID                          2500 non-null   int64  
 1   SUMLEVEL                     2500 non-null   int64  
 2   COUNTYID                     2500 non-null   int64  
 3   STATEID                      2500 non-null   int64  
 4   state                        2500 non-null   object 
 5   state_ab                     2500 non-null   object 
 6   city                         2500 non-null   object 
 7   place                        2500 non-null   object 
 8   type                         2500 non-null   object 
 9   primary                      2500 non-null   object 
 10  zip_code                     2500 non-null   int64  
 11  area_code                    2500 non-null   int64  
 12  lat                          2500 non-null   float64
 13  lng                          2500 non-null   float64
 14  ALand                        2500 non-null   int64  
 15  AWater                       2500 non-null   int64  
 16  pop                          2500 non-null   int64  
 17  male_pop                     2500 non-null   int64  
 18  female_pop                   2500 non-null   int64  
 19  rent_mean                    2500 non-null   float64
 20  rent_median                  2500 non-null   float64
 21  rent_stdev                   2500 non-null   float64
 22  rent_sample_weight           2500 non-null   float64
 23  rent_samples                 2500 non-null   float64
 24  rent_gt_10                   2500 non-null   float64
 25  rent_gt_15                   2500 non-null   float64
 26  rent_gt_20                   2500 non-null   float64
 27  rent_gt_25                   2500 non-null   float64
 28  rent_gt_30                   2500 non-null   float64
 29  rent_gt_35                   2500 non-null   float64
 30  rent_gt_40                   2500 non-null   float64
 31  rent_gt_50                   2500 non-null   float64
 32  universe_samples             2500 non-null   int64  
 33  used_samples                 2500 non-null   int64  
 34  hi_mean                      2500 non-null   float64
 35  hi_median                    2500 non-null   float64
 36  hi_stdev                     2500 non-null   float64
 37  hi_sample_weight             2500 non-null   float64
 38  hi_samples                   2500 non-null   float64
 39  family_mean                  2500 non-null   float64
 40  family_median                2500 non-null   float64
 41  family_stdev                 2500 non-null   float64
 42  family_sample_weight         2500 non-null   float64
 43  family_samples               2500 non-null   float64
 44  hc_mortgage_mean             2500 non-null   float64
 45  hc_mortgage_median           2500 non-null   float64
 46  hc_mortgage_stdev            2500 non-null   float64
 47  hc_mortgage_sample_weight    2500 non-null   float64
 48  hc_mortgage_samples          2500 non-null   float64
 49  hc_mean                      2500 non-null   float64
 50  hc_median                    2500 non-null   float64
 51  hc_stdev                     2500 non-null   float64
 52  hc_samples                   2500 non-null   float64
 53  hc_sample_weight             2500 non-null   float64
 54  home_equity_second_mortgage  2500 non-null   float64
 55  second_mortgage              2500 non-null   float64
 56  home_equity                  2500 non-null   float64
 57  debt                         2500 non-null   float64
 58  second_mortgage_cdf          2500 non-null   float64
 59  home_equity_cdf              2500 non-null   float64
 60  debt_cdf                     2500 non-null   float64
 61  hs_degree                    2500 non-null   float64
 62  hs_degree_male               2500 non-null   float64
 63  hs_degree_female             2500 non-null   float64
 64  male_age_mean                2500 non-null   float64
 65  male_age_median              2500 non-null   float64
 66  male_age_stdev               2500 non-null   float64
 67  male_age_sample_weight       2500 non-null   float64
 68  male_age_samples             2500 non-null   float64
 69  female_age_mean              2500 non-null   float64
 70  female_age_median            2500 non-null   float64
 71  female_age_stdev             2500 non-null   float64
 72  female_age_sample_weight     2500 non-null   float64
```

```
 73  female_age_samples           2500 non-null   float64
 74  pct_own                      2500 non-null   float64
 75  married                      2500 non-null   float64
 76  married_snp                  2500 non-null   float64
 77  separated                    2500 non-null   float64
 78  divorced                     2500 non-null   float64
 79  Bad_debt                     2500 non-null   float64
 80  Good_debt                    2500 non-null   float64
 81  Remaining_income             2500 non-null   float64
 82  Population_density           2500 non-null   float64
 83  median_age                   2500 non-null   float64
 84  rent%                        2500 non-null   float64
dtypes: float64(66), int64(13), object(6)
memory usage: 1.6+ MB
```

In [116…
```
numerical_variables_test = df_test1.select_dtypes(('int64','float64'))
```

In [117…
```
numerical_variables_test.head()
```

Out[117]:

	UID	SUMLEVEL	COUNTYID	STATEID	zip_code	area_code	lat	lng	ALan
6238	266140	140	5	36	10452	718	40.842166	-73.926952	28270
9088	248877	140	33	22	70802	225	30.414676	-91.192011	299069
8771	254689	140	125	26	48075	248	42.453800	-83.207546	148008
4976	252317	140	33	24	20783	301	39.006934	-76.974603	31790
11051	278176	140	101	42	19104	215	39.953811	-75.207043	28354

5 rows × 79 columns

In [118…
```
numerical_variables_test.drop(['SUMLEVEL','lat','lng','ALand','AWater'],axis=1,inplace
```

In [119…
```
numerical_variables_test.shape
```

Out[119]:
```
(2500, 74)
```

In [120…
```
numerical_variables_test.columns
```

Out[120]:
```
Index(['UID', 'COUNTYID', 'STATEID', 'zip_code', 'area_code', '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', 'Remaining_income', 'Population_density',
       'median_age', 'rent%'],
      dtype='object')
```

In [121…
```
X = numerical_variables_test[['UID', 'COUNTYID', 'STATEID', 'zip_code', 'area_code', '
       '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_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', 'Remaining_income', 'Population_density',
        'median_age', 'rent%']]
Y = numerical_variables_test['hc_mortgage_mean']
```

In [122...
```
Y_Pred = lm.predict(X)
```

In [123...
```
MAE1 = mean_absolute_error(Y,Y_Pred)
MAE1
```

Out[123]:
```
47.100968799960974
```

In [124...
```
MSE1 = mean_squared_error(Y,Y_Pred)
MSE1
```

Out[124]:
```
4277.1663000160515
```

In [125...
```
RMSE1 = np.sqrt(MSE1)
RMSE1
```

Out[125]:
```
65.4000481652426
```

In [126...
```
r2_1 = r2_score(Y,Y_Pred)
r2_1
```

Out[126]:
```
0.9875219499655888
```

Here we have 98.75% r2 score so we can skip the state level model building procedure

In [127...
```
Check = pd.DataFrame({'Predicted hc_mortgage_mean' : Y_Pred , 'Actual hc_mortgage_mear
Check
```

Out[127]:

	Predicted hc_mortgage_mean	Actual hc_mortgage_mean
6238	2521.735791	2631.10494
9088	1278.514992	1141.54196
8771	1614.635009	1473.67252
4976	1974.730817	1923.34919
11051	2745.130946	2900.21786
...	...	...
1620	1518.081700	1444.61336
5324	2620.120047	2594.75884
9443	1311.426800	1343.19912
8107	1248.826557	1268.52462
6099	1012.287750	947.51606

2500 rows × 2 columns

In [128...
```
from statsmodels.stats.outliers_influence import variance_inflation_factor
```

In [129...
```
VIF_data = pd.DataFrame()
```

In [130...
```
VIF_data['features'] = numerical_variables_test.columns
```

In [131...
```
VIF_data['VIF'] = [variance_inflation_factor(numerical_variables_test.values,i)
                  for i in range (len(numerical_variables_test.columns))]
```

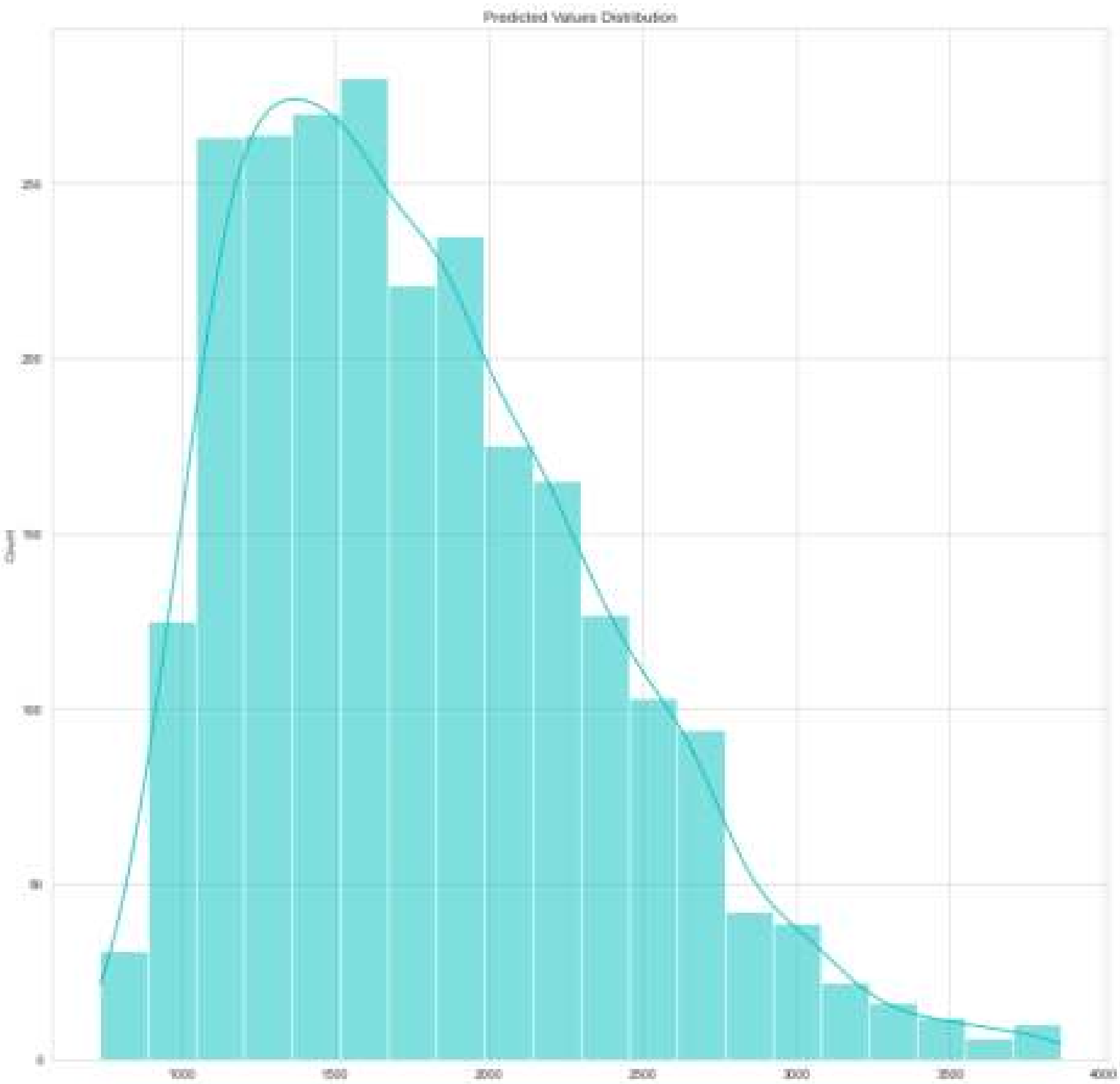
In [132...
```
print(VIF_data)
```

```
                 features          VIF
0                     UID  3542.365500
1                COUNTYID     1.840219
2                 STATEID    89.982089
3                zip_code     6.361545
4               area_code     8.416495
..                    ...          ...
69              Good_debt          inf
70       Remaining_income          inf
71     Population_density     2.673766
72             median_age   346.316974
73                  rent%    64.359182

[74 rows x 2 columns]
```

In [133…

```
plt.figure(figsize=(15,15))
sns.histplot(data=Y_Pred,color='c',bins=20,kde=True)
plt.title('Predicted Values Distribution')
plt.show()
```



The predicted data looks somewhat right skewed

Now we will use pandas function to extract the top 2500 dataframe into csv format for Dashboarding use

In [134…

```
df_train1.to_csv('Real_Estate.csv')
```