In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import norm
```

In [2]:
```
train_data = pd.read_csv(r'C:\Sachin new\Simplilearn\Capstone Project\Real Estate Project\train.csv')
```

In [3]:
```
train_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27321 entries, 0 to 27320
Data columns (total 80 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   UID                         27321 non-null  int64  
 1   BLOCKID                     0 non-null      float64
 2   SUMLEVEL                    27321 non-null  int64  
 3   COUNTYID                    27321 non-null  int64  
 4   STATEID                     27321 non-null  int64  
 5   state                       27321 non-null  object 
 6   state_ab                    27321 non-null  object 
 7   city                        27321 non-null  object 
 8   place                       27321 non-null  object 
 9   type                        27321 non-null  object 
 10  primary                     27321 non-null  object 
 11  zip_code                    27321 non-null  int64  
 12  area_code                   27321 non-null  int64  
 13  lat                         27321 non-null  float64
 14  lng                         27321 non-null  float64
 15  ALand                       27321 non-null  float64
 16  AWater                      27321 non-null  int64  
 17  pop                         27321 non-null  int64  
 18  male_pop                    27321 non-null  int64  
 19  female_pop                  27321 non-null  int64  
 20  rent_mean                   27007 non-null  float64
 21  rent_median                 27007 non-null  float64
 22  rent_stdev                  27007 non-null  float64
 23  rent_sample_weight          27007 non-null  float64
 24  rent_samples                27007 non-null  float64
 25  rent_gt_10                  27007 non-null  float64
 26  rent_gt_15                  27007 non-null  float64
 27  rent_gt_20                  27007 non-null  float64
 28  rent_gt_25                  27007 non-null  float64
 29  rent_gt_30                  27007 non-null  float64
 30  rent_gt_35                  27007 non-null  float64
 31  rent_gt_40                  27007 non-null  float64
 32  rent_gt_50                  27007 non-null  float64
 33  universe_samples            27321 non-null  int64  
 34  used_samples                27321 non-null  int64  
 35  hi_mean                     27053 non-null  float64
 36  hi_median                   27053 non-null  float64
 37  hi_stdev                    27053 non-null  float64
 38  hi_sample_weight            27053 non-null  float64
```

```
 39  hi_samples                   27053 non-null  float64
 40  family_mean                  27023 non-null  float64
 41  family_median                27023 non-null  float64
 42  family_stdev                 27023 non-null  float64
 43  family_sample_weight         27023 non-null  float64
 44  family_samples               27023 non-null  float64
 45  hc_mortgage_mean             26748 non-null  float64
 46  hc_mortgage_median           26748 non-null  float64
 47  hc_mortgage_stdev            26748 non-null  float64
 48  hc_mortgage_sample_weight    26748 non-null  float64
 49  hc_mortgage_samples          26748 non-null  float64
 50  hc_mean                      26721 non-null  float64
 51  hc_median                    26721 non-null  float64
 52  hc_stdev                     26721 non-null  float64
 53  hc_samples                   26721 non-null  float64
 54  hc_sample_weight             26721 non-null  float64
 55  home_equity_second_mortgage  26864 non-null  float64
 56  second_mortgage              26864 non-null  float64
 57  home_equity                  26864 non-null  float64
 58  debt                         26864 non-null  float64
 59  second_mortgage_cdf          26864 non-null  float64
 60  home_equity_cdf              26864 non-null  float64
 61  debt_cdf                     26864 non-null  float64
 62  hs_degree                    27131 non-null  float64
 63  hs_degree_male               27121 non-null  float64
 64  hs_degree_female             27098 non-null  float64
 65  male_age_mean                27132 non-null  float64
 66  male_age_median              27132 non-null  float64
 67  male_age_stdev               27132 non-null  float64
 68  male_age_sample_weight       27132 non-null  float64
 69  male_age_samples             27132 non-null  float64
 70  female_age_mean              27115 non-null  float64
 71  female_age_median            27115 non-null  float64
 72  female_age_stdev             27115 non-null  float64
 73  female_age_sample_weight     27115 non-null  float64
 74  female_age_samples           27115 non-null  float64
 75  pct_own                      27053 non-null  float64
 76  married                      27130 non-null  float64
 77  married_snp                  27130 non-null  float64
 78  separated                    27130 non-null  float64
 79  divorced                     27130 non-null  float64
dtypes: float64(62), int64(12), object(6)
memory usage: 16.7+ MB
```

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

Out[4]:

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

5 rows × 80 columns

Checking and removing duplicate values

In [5]:
```
train_data['UID'].duplicated().sum()
```

Out[5]:
```
160
```

In [6]:
```
train_data.drop_duplicates(inplace = True)
```

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

Out[7]:
```
(27161, 80)
```

In [8]:
```
pd.set_option('display.max_rows', 100)
pd.set_option('display.max_columns',None)
```

Checking UID has all unique values. If yes, then setting UID as Index

In [9]:
```
train_data['UID'].nunique()
```

Out[9]:
```
27161
```

In [10]:
```
train_data.set_index('UID',inplace = True)
```

In [11]:
```
train_data.isna().sum()
```

Out[11]:

BLOCKID	27161
SUMLEVEL	0
COUNTYID	0
STATEID	0
state	0
state_ab	0
city	0
place	0
type	0
primary	0
zip_code	0
area_code	0
lat	0
lng	0
ALand	0
AWater	0
pop	0
male_pop	0
female_pop	0
rent_mean	242
rent_median	242
rent_stdev	242
rent_sample_weight	242
rent_samples	242
rent_gt_10	242
rent_gt_15	242
rent_gt_20	242
rent_gt_25	242
rent_gt_30	242
rent_gt_35	242
rent_gt_40	242
rent_gt_50	242
universe_samples	0
used_samples	0
hi_mean	207
hi_median	207
hi_stdev	207
hi_sample_weight	207
hi_samples	207
family_mean	230
family_median	230
family_stdev	230
family_sample_weight	230
family_samples	230

```
hc_mortgage_mean                442
hc_mortgage_median              442
hc_mortgage_stdev               442
hc_mortgage_sample_weight       442
hc_mortgage_samples             442
hc_mean                         478
hc_median                       478
hc_stdev                        478
hc_samples                      478
hc_sample_weight                478
home_equity_second_mortgage     360
second_mortgage                 360
home_equity                     360
debt                            360
second_mortgage_cdf             360
home_equity_cdf                 360
debt_cdf                        360
hs_degree                       145
hs_degree_male                  154
hs_degree_female                171
male_age_mean                   148
male_age_median                 148
male_age_stdev                  148
male_age_sample_weight          148
male_age_samples                148
female_age_mean                 161
female_age_median               161
female_age_stdev                161
female_age_sample_weight        161
female_age_samples              161
pct_own                         207
married                         150
married_snp                     150
separated                       150
divorced                        150
dtype: int64
```

In [12]:
```
pd.DataFrame({'Missing Count': train_data.isna().sum(), 'Missing Percent': (train_data.isna().mean().round(4)*100)})
```

Out[12]:

	Missing Count	Missing Percent
BLOCKID	27161	100.00
SUMLEVEL	0	0.00
COUNTYID	0	0.00
STATEID	0	0.00
state	0	0.00
state_ab	0	0.00
city	0	0.00
place	0	0.00
type	0	0.00
primary	0	0.00
zip_code	0	0.00
area_code	0	0.00
lat	0	0.00
lng	0	0.00
ALand	0	0.00
AWater	0	0.00
pop	0	0.00
male_pop	0	0.00
female_pop	0	0.00
rent_mean	242	0.89
rent_median	242	0.89
rent_stdev	242	0.89
rent_sample_weight	242	0.89
rent_samples	242	0.89

	Missing Count	Missing Percent
rent_gt_10	242	0.89
rent_gt_15	242	0.89
rent_gt_20	242	0.89
rent_gt_25	242	0.89
rent_gt_30	242	0.89
rent_gt_35	242	0.89
rent_gt_40	242	0.89
rent_gt_50	242	0.89
universe_samples	0	0.00
used_samples	0	0.00
hi_mean	207	0.76
hi_median	207	0.76
hi_stdev	207	0.76
hi_sample_weight	207	0.76
hi_samples	207	0.76
family_mean	230	0.85
family_median	230	0.85
family_stdev	230	0.85
family_sample_weight	230	0.85
family_samples	230	0.85
hc_mortgage_mean	442	1.63
hc_mortgage_median	442	1.63
hc_mortgage_stdev	442	1.63
hc_mortgage_sample_weight	442	1.63

	Missing Count	Missing Percent
hc_mortgage_samples	442	1.63
hc_mean	478	1.76
hc_median	478	1.76
hc_stdev	478	1.76
hc_samples	478	1.76
hc_sample_weight	478	1.76
home_equity_second_mortgage	360	1.33
second_mortgage	360	1.33
home_equity	360	1.33
debt	360	1.33
second_mortgage_cdf	360	1.33
home_equity_cdf	360	1.33
debt_cdf	360	1.33
hs_degree	145	0.53
hs_degree_male	154	0.57
hs_degree_female	171	0.63
male_age_mean	148	0.54
male_age_median	148	0.54
male_age_stdev	148	0.54
male_age_sample_weight	148	0.54
male_age_samples	148	0.54
female_age_mean	161	0.59
female_age_median	161	0.59
female_age_stdev	161	0.59

	Missing Count	Missing Percent
female_age_sample_weight	161	0.59
female_age_samples	161	0.59
pct_own	207	0.76
married	150	0.55
married_snp	150	0.55
separated	150	0.55
divorced	150	0.55

Remove BlockID since it is completely blank

In [13]:
```
train_data.drop(['BLOCKID'], axis = 1, inplace = True)
```

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

Out[14]:

UID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	type	primary	zip_code	area_code	lat	lng	AL
267822	140	53	36	New York	NY	Hamilton	Hamilton	City	tract	13346	315	42.840812	-75.501524	2021833
246444	140	141	18	Indiana	IN	South Bend	Roseland	City	tract	46616	574	41.701441	-86.266614	15608
245683	140	63	18	Indiana	IN	Danville	Danville	City	tract	46122	317	39.792202	-86.515246	695615
279653	140	127	72	Puerto Rico	PR	San Juan	Guaynabo	Urban	tract	927	787	18.396103	-66.104169	11057
247218	140	161	20	Kansas	KS	Manhattan	Manhattan City	City	tract	66502	785	39.195573	-96.569366	25544

Remove 'Primary' column and 'Sumlevel' column, since they have the same values

In [15]:
```
train_data.drop(train_data[['primary','SUMLEVEL']], axis = 1, inplace=True)
```

Checking how many rows have population = Zero. What are the other data in those rows

In [16]:
```
(train_data['pop']==0).value_counts()
```

Out[16]:
```
False    27019
True       142
Name: pop, dtype: int64
```

In [17]:
```
train_data[train_data['pop']==0]
```

Out[17]:

UID	COUNTYID	STATEID	state	state_ab	city	place	type	zip_code	area_code	lat	lng	ALand	AWater
267158	47	36	New York	NY	Brooklyn	New York City	City	11215	718	40.659126	-73.969773	2313042.0	227326
290217	670	51	Virginia	VA	Hopewell	Hopewell City	Town	23860	804	37.289420	-77.269743	5940771.0	10627
247072	91	20	Kansas	KS	Shawnee Mission	Lenexa City	City	66219	913	38.937953	-94.768942	13379076.0	148768
253670	65	26	Michigan	MI	Lansing	Edgemont Park	CDP	48915	517	42.739759	-84.580693	3105697.0	289337
262696	15	33	New Hampshire	NH	Manchester	Manchester City	CDP	3103	603	42.924101	-71.435913	3827015.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
268317	61	36	New York	NY	New York	New York City	City	10004	212	40.690055	-74.046142	78638.0	0
250649	17	25	Massachusetts	MA	Maynard	Maynard	City	1754	978	42.409915	-71.481881	3842441.0	16212
230726	81	6	California	CA	Burlingame	Millbrae City	City	94019	650	37.614777	-122.376913	7920265.0	3591281
293176	79	55	Wisconsin	WI	Milwaukee	Shorewood	City	53202	414	43.046856	-87.889666	2062620.0	3971772
275500	51	41	Oregon	OR	Portland	Portland City	City	97217	503	45.561643	-122.706202	4078535.0	1594068

142 rows × 76 columns

Since all rows, where population is zero, the other columns are also blank. Hence deleting those rows

In [18]:
```
train_data = train_data.drop(train_data[train_data['pop']==0].index)
```

In [19]:
```
pd.DataFrame({'Missing Count': train_data.isna().sum(), 'Missing Percent': (train_data.isna().mean().round(4)*100)})
```

Out[19]:

	Missing Count	Missing Percent
COUNTYID	0	0.00
STATEID	0	0.00
state	0	0.00
state_ab	0	0.00
city	0	0.00
place	0	0.00
type	0	0.00
zip_code	0	0.00
area_code	0	0.00
lat	0	0.00
lng	0	0.00
ALand	0	0.00
AWater	0	0.00
pop	0	0.00
male_pop	0	0.00
female_pop	0	0.00
rent_mean	100	0.37
rent_median	100	0.37
rent_stdev	100	0.37
rent_sample_weight	100	0.37
rent_samples	100	0.37
rent_gt_10	100	0.37
rent_gt_15	100	0.37
rent_gt_20	100	0.37

	Missing Count	Missing Percent
rent_gt_25	100	0.37
rent_gt_30	100	0.37
rent_gt_35	100	0.37
rent_gt_40	100	0.37
rent_gt_50	100	0.37
universe_samples	0	0.00
used_samples	0	0.00
hi_mean	65	0.24
hi_median	65	0.24
hi_stdev	65	0.24
hi_sample_weight	65	0.24
hi_samples	65	0.24
family_mean	88	0.33
family_median	88	0.33
family_stdev	88	0.33
family_sample_weight	88	0.33
family_samples	88	0.33
hc_mortgage_mean	300	1.11
hc_mortgage_median	300	1.11
hc_mortgage_stdev	300	1.11
hc_mortgage_sample_weight	300	1.11
hc_mortgage_samples	300	1.11
hc_mean	336	1.24
hc_median	336	1.24

	Missing Count	Missing Percent
hc_stdev	336	1.24
hc_samples	336	1.24
hc_sample_weight	336	1.24
home_equity_second_mortgage	218	0.81
second_mortgage	218	0.81
home_equity	218	0.81
debt	218	0.81
second_mortgage_cdf	218	0.81
home_equity_cdf	218	0.81
debt_cdf	218	0.81
hs_degree	3	0.01
hs_degree_male	12	0.04
hs_degree_female	29	0.11
male_age_mean	6	0.02
male_age_median	6	0.02
male_age_stdev	6	0.02
male_age_sample_weight	6	0.02
male_age_samples	6	0.02
female_age_mean	19	0.07
female_age_median	19	0.07
female_age_stdev	19	0.07
female_age_sample_weight	19	0.07
female_age_samples	19	0.07
pct_own	65	0.24

	Missing Count	Missing Percent
married	8	0.03
married_snp	8	0.03
separated	8	0.03
divorced	8	0.03

Checking how many rows have no family income details. What are the other key data in those rows

In [20]:
```
train_data[train_data['hi_mean'].isna()]
```

Out[20]:

UID	COUNTYID	STATEID	state	state_ab	city	place	type	zip_code	area_code	lat	lng	ALand
223593	19	4	Arizona	AZ	Tucson	Littletown	CDP	85734	520	32.067721	-110.867177	2909152.0
246589	157	18	Indiana	IN	West Lafayette	West Lafayette City	City	47906	765	40.427960	-86.925357	771959.0
286305	245	48	Texas	TX	Port Arthur	Central Gardens	Town	77641	409	29.985982	-94.050079	3500203.0
239655	121	13	Georgia	GA	Atlanta	Gresham Park	City	30315	404	33.710292	-84.367788	660642.0
262260	109	31	Nebraska	NE	Lincoln	Yankee Hill	Village	68512	402	40.770017	-96.703843	194262.0
279189	37	72	Puerto Rico	PR	Ceiba	Ceiba	Urban	735	787	18.238828	-65.626836	31232980.0
252510	510	24	Maryland	MD	Baltimore	Baltimore City	CDP	21202	410	39.299511	-76.609125	156309.0
231894	111	6	California	CA	Avalon	Channel Islands Beach	City	90704	310	33.255655	-119.503588	58634146.0
227654	53	6	California	CA	Monterey	Marina City	City	93942	831	36.651862	-121.798753	3204492.0
282619	157	47	Tennessee	TN	Memphis	Bartlett City	City	38134	901	35.156950	-89.867071	2277941.0
222533	13	4	Arizona	AZ	Goodyear	Citrus Park	CDP	85338	623	33.471162	-112.438036	3244255.0
225038	31	6	California	CA	Corcoran	Corcoran City	City	93212	559	36.057953	-119.554067	5116440.0
282677	157	47	Tennessee	TN	Memphis	Memphis City	City	38107	901	35.053396	-89.971244	19126957.0
264053	25	34	New Jersey	NJ	Tinton Falls	Eatontown	City	7724	732	40.296594	-74.079991	2225754.0
290423	740	51	Virginia	VA	Portsmouth	Portsmouth City	Town	23704	757	36.811594	-76.296828	2957280.0
253628	65	26	Michigan	MI	East Lansing	East Lansing City	CDP	48823	517	42.730749	-84.473513	88527.0
254290	99	26	Michigan	MI	Warren	Warren City	CDP	48093	586	42.518869	-83.035638	3590129.0
251077	25	25	Massachusetts	MA	Hull	Hull	City	2045	781	42.326512	-70.939034	3270315.0
273184	129	39	Ohio	OH	Grove City	Orient	Village	43123	614	39.794978	-83.148196	5147772.0
260311	77	37	North Carolina	NC	Butner	Butner	Village	27509	919	36.154845	-78.793625	7967277.0

UID	COUNTYID	STATEID	state	state_ab	city	place	type	zip_code	area_code	lat	lng	ALand
232591	41	8	Colorado	CO	Usaf Academy	Air Force Academy	City	80840	719	39.007297	-104.852302	48509858.0
253961	87	26	Michigan	MI	Lapeer	Lapeer City	CDP	48446	810	43.034784	-83.349058	309085.0
266664	27	36	New York	NY	Poughquag	Hopewell Junction	City	12570	845	41.578838	-73.722796	3206552.0
289640	121	51	Virginia	VA	Blacksburg	Blacksburg	Town	24060	540	37.224746	-80.427121	3306357.0
267846	55	36	New York	NY	Rochester	Brighton	City	14611	585	43.128664	-77.629144	398890.0
249424	71	22	Louisiana	LA	New Orleans	Jefferson	City	70119	504	29.960629	-90.095006	206718.0
233994	13	9	Connecticut	CT	Somers	Hazardville	CDP	6071	860	42.020792	-72.500163	1609529.0
263271	11	34	New Jersey	NJ	Bridgeton	Bridgeton City	City	8302	856	39.413214	-75.207604	353642.0
247777	61	21	Kentucky	KY	Brownsville	Brownsville City	City	42210	270	37.197158	-86.156329	175916489.0
251083	25	25	Massachusetts	MA	South Boston	Boston City	City	2127	617	42.346193	-71.026227	1696453.0
261031	133	37	North Carolina	NC	Camp Lejuene	Jacksonville City	Village	28547	910	34.664168	-77.342056	19141765.0
221306	97	1	Alabama	AL	Mobile	Mobile City	Town	36608	251	30.695143	-88.182485	1900631.0
257114	163	27	Minnesota	MN	Bayport	Bayport City	City	55003	651	45.027390	-92.789324	215546.0
288917	25	51	Virginia	VA	Lawrenceville	Lawrenceville	Town	23868	434	36.782642	-77.819836	2193311.0
266866	29	36	New York	NY	Alden	Town Line	City	14004	716	42.928389	-78.550754	3568791.0
266722	29	36	New York	NY	Buffalo	Kenmore	City	14207	716	42.934270	-78.883564	456373.0
287648	453	48	Texas	TX	Austin	Austin City	Town	78703	512	30.305673	-97.760829	503773.0
236972	86	12	Florida	FL	Miami	Fisher Island	City	33132	305	25.775079	-80.167092	2616068.0
224998	29	6	California	CA	Tehachapi	Stallion Springs	City	93561	661	35.112906	-118.570979	6754420.0
224970	29	6	California	CA	Delano	Delano City	City	93215	661	35.767313	-119.322309	1958572.0

UID	COUNTYID	STATEID	state	state_ab	city	place	type	zip_code	area_code	lat	lng	ALand
260069	63	37	North Carolina	NC	Durham	Durham City	Village	27701	919	36.006844	-78.915669	419510.0
251360	3	24	Maryland	MD	Jessup	Jessup	CDP	20794	410	39.135461	-76.777831	2705362.0
263780	21	34	New Jersey	NJ	Trenton	Trenton City	City	8611	609	40.207185	-74.756664	41134.0
281635	37	47	Tennessee	TN	Nashville	Nashville-davidson Metropolitan Government	City	37209	615	36.179041	-86.883456	14833107.0
282028	93	47	Tennessee	TN	Knoxville	Knoxville City	City	37902	865	35.954209	-83.924555	690041.0
257115	163	27	Minnesota	MN	Stillwater	Oak Park Heights City	City	55082	651	45.024958	-92.802094	738408.0
268309	59	36	New York	NY	Garden City	Garden City	City	11530	516	40.720501	-73.652072	273322.0
231704	107	6	California	CA	Porterville	East Porterville	City	93257	559	36.038061	-118.983849	1272840.0
268459	61	36	New York	NY	New York	New York City	City	10028	212	40.781317	-73.966847	2991654.0
266490	15	36	New York	NY	Elmira	Elmira Heights	City	14901	607	42.112845	-76.832349	480229.0
268311	59	36	New York	NY	Jericho	Brookville	City	11753	516	40.793408	-73.570256	2799804.0
227472	41	6	California	CA	Corte Madera	Corte Madera	City	94925	415	37.940598	-122.490688	140958.0
252424	43	24	Maryland	MD	Hagerstown	Breathedsville	CDP	21740	301	39.559686	-77.714795	4425975.0
276837	41	42	Pennsylvania	PA	Camp Hill	Lower Allen	Borough	17011	717	40.213480	-76.920378	2434052.0
238950	53	13	Georgia	GA	Cusseta	Cusseta-chattahoochee	City	31805	706	32.418280	-84.750912	303578182.0
224778	19	6	California	CA	Coalinga	Coalinga City	City	93210	559	36.130045	-120.244927	2619931.0
232683	43	8	Colorado	CO	Florence	Florence City	City	81226	719	38.359531	-105.097939	2582577.0
263500	13	34	New Jersey	NJ	Newark	Newark City	City	7105	973	40.696737	-74.156884	16267707.0
236899	86	12	Florida	FL	Miami	Tamiami	City	33175	305	25.777366	-80.425988	35057502.0

UID	COUNTYID	STATEID	state	state_ab	city	place	type	zip_code	area_code	lat	lng	ALand
269337	81	36	New York	NY	Saint Albans	North Valley Stream	City	11412	718	40.688870	-73.770230	524647.0
281472	9	47	Tennessee	TN	Alcoa	Alcoa City	City	37701	865	35.809206	-83.996679	6365439.0
288420	35	49	Utah	UT	Draper	Draper City	City	84020	801	40.495276	-111.901391	3099460.0
229543	71	6	California	CA	Highland	Highland City	City	92346	909	34.139018	-117.219758	857381.0
227390	37	6	California	CA	Los Angeles	Burbank City	City	90027	323	34.127607	-118.296387	20912226.0

In [21]:
```
train_data[train_data['hi_mean'].isna()].shape
```

Out[21]:
```
(65, 76)
```

Since these rows also have missing information for other key features such as Rent, Mortgage cost, home owner cost, loan/debt details, hence these are also deleted

In [22]:
```
train_data = train_data.drop(train_data[train_data['hi_mean'].isna()].index)
```

In [23]:
```
pd.DataFrame({'Missing Count': train_data.isna().sum(), 'Missing Percent': (train_data.isna().mean().round(4)*100)})
```

Out[23]:

	Missing Count	Missing Percent
COUNTYID	0	0.00
STATEID	0	0.00
state	0	0.00
state_ab	0	0.00
city	0	0.00
place	0	0.00
type	0	0.00
zip_code	0	0.00
area_code	0	0.00
lat	0	0.00
lng	0	0.00
ALand	0	0.00
AWater	0	0.00
pop	0	0.00
male_pop	0	0.00
female_pop	0	0.00
rent_mean	35	0.13
rent_median	35	0.13
rent_stdev	35	0.13
rent_sample_weight	35	0.13
rent_samples	35	0.13
rent_gt_10	35	0.13
rent_gt_15	35	0.13
rent_gt_20	35	0.13

	Missing Count	Missing Percent
rent_gt_25	35	0.13
rent_gt_30	35	0.13
rent_gt_35	35	0.13
rent_gt_40	35	0.13
rent_gt_50	35	0.13
universe_samples	0	0.00
used_samples	0	0.00
hi_mean	0	0.00
hi_median	0	0.00
hi_stdev	0	0.00
hi_sample_weight	0	0.00
hi_samples	0	0.00
family_mean	23	0.09
family_median	23	0.09
family_stdev	23	0.09
family_sample_weight	23	0.09
family_samples	23	0.09
hc_mortgage_mean	235	0.87
hc_mortgage_median	235	0.87
hc_mortgage_stdev	235	0.87
hc_mortgage_sample_weight	235	0.87
hc_mortgage_samples	235	0.87
hc_mean	271	1.01
hc_median	271	1.01

	Missing Count	Missing Percent
hc_stdev	271	1.01
hc_samples	271	1.01
hc_sample_weight	271	1.01
home_equity_second_mortgage	153	0.57
second_mortgage	153	0.57
home_equity	153	0.57
debt	153	0.57
second_mortgage_cdf	153	0.57
home_equity_cdf	153	0.57
debt_cdf	153	0.57
hs_degree	1	0.00
hs_degree_male	7	0.03
hs_degree_female	5	0.02
male_age_mean	4	0.01
male_age_median	4	0.01
male_age_stdev	4	0.01
male_age_sample_weight	4	0.01
male_age_samples	4	0.01
female_age_mean	4	0.01
female_age_median	4	0.01
female_age_stdev	4	0.01
female_age_sample_weight	4	0.01
female_age_samples	4	0.01
pct_own	0	0.00

	Missing Count	Missing Percent
married	6	0.02
married_snp	6	0.02
separated	6	0.02
divorced	6	0.02

In [24]:
```
train_data.shape
```

Out[24]:
```
(26954, 76)
```

There are still missing cases w.r.t. rental information, household income, family income, monthly mortgage and owners cost. Since these are related to the geographic location, the missing values are filled with average values for the respective "state"

In [25]:
```
train_data.columns
```

Out[25]:
```
Index(['COUNTYID', 'STATEID', 'state', 'state_ab', 'city', 'place', 'type',
       'zip_code', 'area_code', 'lat', 'lng', 'ALand', 'AWater', 'pop',
       'male_pop', 'female_pop', 'rent_mean', 'rent_median', 'rent_stdev',
       'rent_sample_weight', 'rent_samples', 'rent_gt_10', 'rent_gt_15',
       'rent_gt_20', 'rent_gt_25', 'rent_gt_30', 'rent_gt_35', 'rent_gt_40',
       'rent_gt_50', 'universe_samples', 'used_samples', 'hi_mean',
       'hi_median', 'hi_stdev', 'hi_sample_weight', 'hi_samples',
       'family_mean', 'family_median', 'family_stdev', 'family_sample_weight',
       'family_samples', 'hc_mortgage_mean', 'hc_mortgage_median',
       'hc_mortgage_stdev', 'hc_mortgage_sample_weight', 'hc_mortgage_samples',
       'hc_mean', 'hc_median', 'hc_stdev', 'hc_samples', 'hc_sample_weight',
       'home_equity_second_mortgage', 'second_mortgage', 'home_equity', 'debt',
       'second_mortgage_cdf', 'home_equity_cdf', 'debt_cdf', 'hs_degree',
       'hs_degree_male', 'hs_degree_female', 'male_age_mean',
       'male_age_median', 'male_age_stdev', 'male_age_sample_weight',
       'male_age_samples', 'female_age_mean', 'female_age_median',
       'female_age_stdev', 'female_age_sample_weight', 'female_age_samples',
       'pct_own', 'married', 'married_snp', 'separated', 'divorced'],
      dtype='object')
```

In [26]:
```
rent_na = ['rent_mean', 'rent_median','rent_stdev', 'rent_sample_weight', 'rent_samples', 'rent_gt_10','rent_gt_15', 'rent_gt_20
       'rent_gt_40', 'rent_gt_50']
```

```
family_na = ['family_mean', 'family_median', 'family_stdev', 'family_sample_weight', 'family_samples']
hc_mortgage_na = ['hc_mortgage_mean', 'hc_mortgage_median', 'hc_mortgage_stdev', 'hc_mortgage_sample_weight', 'hc_mortgage_sample
hc_na = ['hc_mean', 'hc_median', 'hc_stdev', 'hc_samples', 'hc_sample_weight']
```

In [27]:
```
train_data[rent_na] = train_data.groupby('state')[rent_na].transform(lambda x: x.fillna(x.mean()))
train_data[family_na] = train_data.groupby('state')[family_na].transform(lambda x: x.fillna(x.mean()))
train_data[hc_mortgage_na] = train_data.groupby('state')[hc_mortgage_na].transform(lambda x: x.fillna(x.mean()))
train_data[hc_na] = train_data.groupby('state')[hc_na].transform(lambda x: x.fillna(x.mean()))
```

In [28]:
```
pd.DataFrame({'Missing Count': train_data.isna().sum(), 'Missing Percent': (train_data.isna().mean().round(4)*100)})
```

Out[28]:

	Missing Count	Missing Percent
COUNTYID	0	0.00
STATEID	0	0.00
state	0	0.00
state_ab	0	0.00
city	0	0.00
place	0	0.00
type	0	0.00
zip_code	0	0.00
area_code	0	0.00
lat	0	0.00
lng	0	0.00
ALand	0	0.00
AWater	0	0.00
pop	0	0.00
male_pop	0	0.00
female_pop	0	0.00
rent_mean	0	0.00
rent_median	0	0.00
rent_stdev	0	0.00
rent_sample_weight	0	0.00
rent_samples	0	0.00
rent_gt_10	0	0.00
rent_gt_15	0	0.00
rent_gt_20	0	0.00

	Missing Count	Missing Percent
rent_gt_25	0	0.00
rent_gt_30	0	0.00
rent_gt_35	0	0.00
rent_gt_40	0	0.00
rent_gt_50	0	0.00
universe_samples	0	0.00
used_samples	0	0.00
hi_mean	0	0.00
hi_median	0	0.00
hi_stdev	0	0.00
hi_sample_weight	0	0.00
hi_samples	0	0.00
family_mean	0	0.00
family_median	0	0.00
family_stdev	0	0.00
family_sample_weight	0	0.00
family_samples	0	0.00
hc_mortgage_mean	0	0.00
hc_mortgage_median	0	0.00
hc_mortgage_stdev	0	0.00
hc_mortgage_sample_weight	0	0.00
hc_mortgage_samples	0	0.00
hc_mean	0	0.00
hc_median	0	0.00

	Missing Count	Missing Percent
hc_stdev	0	0.00
hc_samples	0	0.00
hc_sample_weight	0	0.00
home_equity_second_mortgage	153	0.57
second_mortgage	153	0.57
home_equity	153	0.57
debt	153	0.57
second_mortgage_cdf	153	0.57
home_equity_cdf	153	0.57
debt_cdf	153	0.57
hs_degree	1	0.00
hs_degree_male	7	0.03
hs_degree_female	5	0.02
male_age_mean	4	0.01
male_age_median	4	0.01
male_age_stdev	4	0.01
male_age_sample_weight	4	0.01
male_age_samples	4	0.01
female_age_mean	4	0.01
female_age_median	4	0.01
female_age_stdev	4	0.01
female_age_sample_weight	4	0.01
female_age_samples	4	0.01
pct_own	0	0.00

	Missing Count	Missing Percent
married	6	0.02
married_snp	6	0.02
separated	6	0.02
divorced	6	0.02

In [29]:
```
train_data.shape
```

Out[29]:
```
(26954, 76)
```

The remaining missing values are filled with the average of the respective columns

In [30]:
```
train_data.fillna(train_data.mean(numeric_only = True), inplace = True)
```

In [31]:
```
pd.DataFrame({'Missing Count': train_data.isna().sum(), 'Missing Percent': (train_data.isna().mean().round(4)*100)})
```

Out[31]:

	Missing Count	Missing Percent
COUNTYID	0	0.0
STATEID	0	0.0
state	0	0.0
state_ab	0	0.0
city	0	0.0
place	0	0.0
type	0	0.0
zip_code	0	0.0
area_code	0	0.0
lat	0	0.0
lng	0	0.0
ALand	0	0.0
AWater	0	0.0
pop	0	0.0
male_pop	0	0.0
female_pop	0	0.0
rent_mean	0	0.0
rent_median	0	0.0
rent_stdev	0	0.0
rent_sample_weight	0	0.0
rent_samples	0	0.0
rent_gt_10	0	0.0
rent_gt_15	0	0.0
rent_gt_20	0	0.0

	Missing Count	Missing Percent
rent_gt_25	0	0.0
rent_gt_30	0	0.0
rent_gt_35	0	0.0
rent_gt_40	0	0.0
rent_gt_50	0	0.0
universe_samples	0	0.0
used_samples	0	0.0
hi_mean	0	0.0
hi_median	0	0.0
hi_stdev	0	0.0
hi_sample_weight	0	0.0
hi_samples	0	0.0
family_mean	0	0.0
family_median	0	0.0
family_stdev	0	0.0
family_sample_weight	0	0.0
family_samples	0	0.0
hc_mortgage_mean	0	0.0
hc_mortgage_median	0	0.0
hc_mortgage_stdev	0	0.0
hc_mortgage_sample_weight	0	0.0
hc_mortgage_samples	0	0.0
hc_mean	0	0.0
hc_median	0	0.0

	Missing Count	Missing Percent
hc_stdev	0	0.0
hc_samples	0	0.0
hc_sample_weight	0	0.0
home_equity_second_mortgage	0	0.0
second_mortgage	0	0.0
home_equity	0	0.0
debt	0	0.0
second_mortgage_cdf	0	0.0
home_equity_cdf	0	0.0
debt_cdf	0	0.0
hs_degree	0	0.0
hs_degree_male	0	0.0
hs_degree_female	0	0.0
male_age_mean	0	0.0
male_age_median	0	0.0
male_age_stdev	0	0.0
male_age_sample_weight	0	0.0
male_age_samples	0	0.0
female_age_mean	0	0.0
female_age_median	0	0.0
female_age_stdev	0	0.0
female_age_sample_weight	0	0.0
female_age_samples	0	0.0
pct_own	0	0.0

	Missing Count	Missing Percent
married	0	0.0
married_snp	0	0.0
separated	0	0.0
divorced	0	0.0

In [32]:
```
#train_data.to_csv('C:\\Sachin new\\Simplilearn\\Capstone Project\\Real Estate Project\\train_updated_25Jan.csv')
```

In [33]:
```
train_data.shape
```

Out[33]:
```
(26954, 76)
```

In [34]:
```
train_data.columns
```

Out[34]:
```
Index(['COUNTYID', 'STATEID', 'state', 'state_ab', 'city', 'place', 'type',
       'zip_code', 'area_code', 'lat', 'lng', 'ALand', 'AWater', 'pop',
       'male_pop', 'female_pop', 'rent_mean', 'rent_median', 'rent_stdev',
       'rent_sample_weight', 'rent_samples', 'rent_gt_10', 'rent_gt_15',
       'rent_gt_20', 'rent_gt_25', 'rent_gt_30', 'rent_gt_35', 'rent_gt_40',
       'rent_gt_50', 'universe_samples', 'used_samples', 'hi_mean',
       'hi_median', 'hi_stdev', 'hi_sample_weight', 'hi_samples',
       'family_mean', 'family_median', 'family_stdev', 'family_sample_weight',
       'family_samples', 'hc_mortgage_mean', 'hc_mortgage_median',
       'hc_mortgage_stdev', 'hc_mortgage_sample_weight', 'hc_mortgage_samples',
       'hc_mean', 'hc_median', 'hc_stdev', 'hc_samples', 'hc_sample_weight',
       'home_equity_second_mortgage', 'second_mortgage', 'home_equity', 'debt',
       'second_mortgage_cdf', 'home_equity_cdf', 'debt_cdf', 'hs_degree',
       'hs_degree_male', 'hs_degree_female', 'male_age_mean',
       'male_age_median', 'male_age_stdev', 'male_age_sample_weight',
       'male_age_samples', 'female_age_mean', 'female_age_median',
       'female_age_stdev', 'female_age_sample_weight', 'female_age_samples',
       'pct_own', 'married', 'married_snp', 'separated', 'divorced'],
      dtype='object')
```

Selecting important features for doing correlation analysis

In [35]:
```
train_correl = train_data[['COUNTYID', 'STATEID','pop','rent_mean','hi_mean','family_mean','hc_mortgage_mean',
                           'hc_mean','home_equity_second_mortgage', 'second_mortgage', 'home_equity', 'debt',
```

```
                                'male_age_mean','female_age_mean','pct_own', 'married', 'separated', 'divorced']]
```

In [36]:
```
train_correl.shape
```

Out[36]:
```
(26954, 18)
```

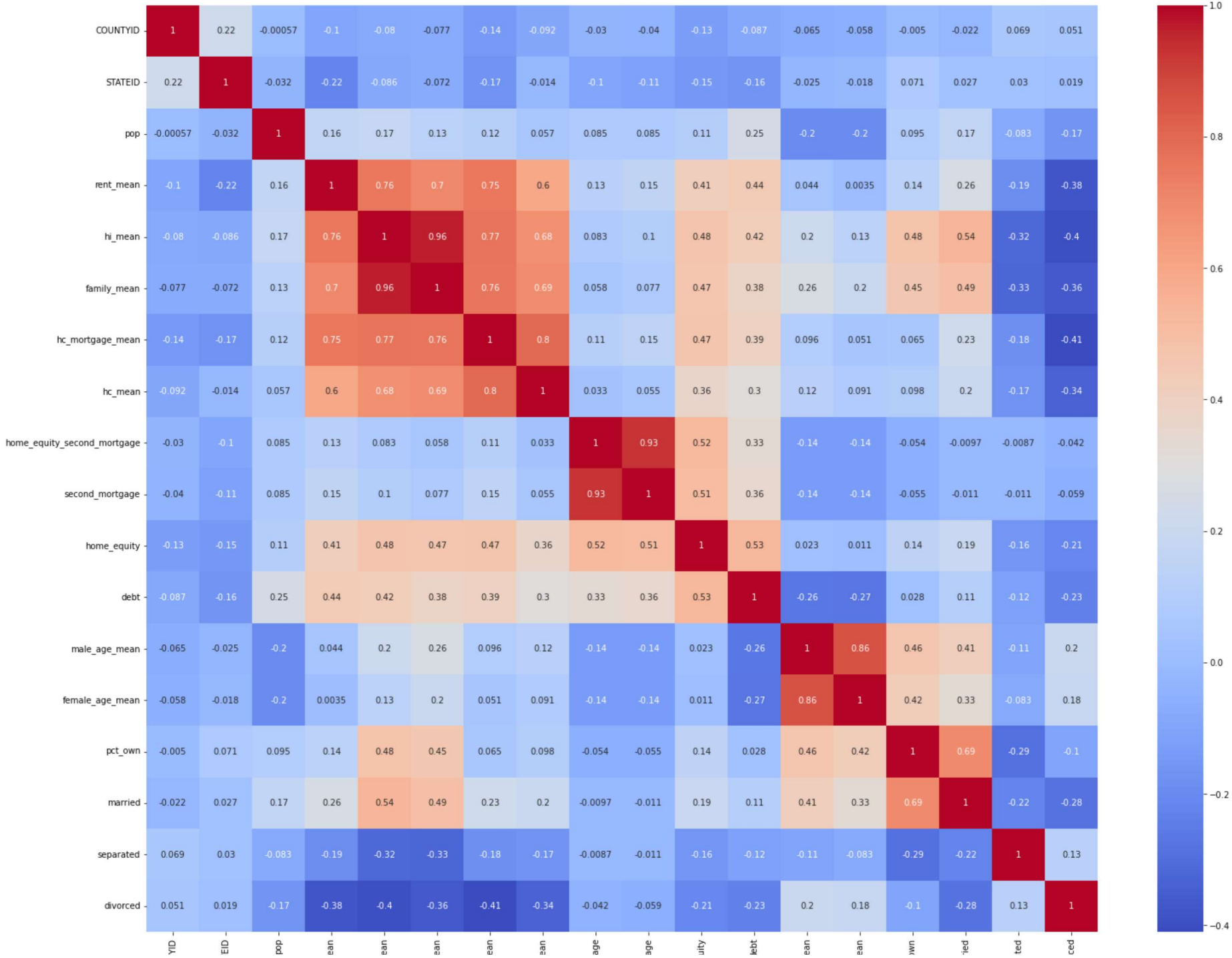
In [37]:
```
# train_correl.to_csv('C:\\Sachin new\\Simplilearn\\Capstone Project\\Real Estate Project\\train_correl.csv')
```

In [38]:
```
train_correl.corr()
```

Out[38]:

	COUNTYID	STATEID	pop	rent_mean	hi_mean	family_mean	hc_mortgage_mean	hc_mean	home_equity_second
COUNTYID	1.000000	0.224565	-0.000567	-0.100521	-0.079543	-0.076742	-0.141659	-0.092458	
STATEID	0.224565	1.000000	-0.032491	-0.217702	-0.086304	-0.072161	-0.169282	-0.014386	
pop	-0.000567	-0.032491	1.000000	0.163491	0.174299	0.134765	0.115557	0.056816	
rent_mean	-0.100521	-0.217702	0.163491	1.000000	0.755850	0.702992	0.753801	0.596587	
hi_mean	-0.079543	-0.086304	0.174299	0.755850	1.000000	0.961763	0.765860	0.675850	
family_mean	-0.076742	-0.072161	0.134765	0.702992	0.961763	1.000000	0.761065	0.688063	
hc_mortgage_mean	-0.141659	-0.169282	0.115557	0.753801	0.765860	0.761065	1.000000	0.796802	
hc_mean	-0.092458	-0.014386	0.056816	0.596587	0.675850	0.688063	0.796802	1.000000	
home_equity_second_mortgage	-0.030399	-0.101082	0.085189	0.129229	0.082510	0.058287	0.114534	0.033088	
second_mortgage	-0.040276	-0.114382	0.085077	0.153189	0.101047	0.077058	0.148739	0.054997	
home_equity	-0.126109	-0.146774	0.105959	0.413234	0.476613	0.465788	0.469283	0.361045	
debt	-0.087372	-0.163281	0.248503	0.438072	0.422640	0.381350	0.392535	0.300787	
male_age_mean	-0.065357	-0.024776	-0.195106	0.043815	0.204455	0.262425	0.096336	0.124545	
female_age_mean	-0.058310	-0.018402	-0.200166	0.003500	0.132075	0.197964	0.051024	0.091350	
pct_own	-0.004983	0.070931	0.095063	0.140578	0.481080	0.451157	0.065044	0.097519	
married	-0.021784	0.026900	0.167639	0.258521	0.538640	0.488259	0.228236	0.201218	
separated	0.068649	0.029564	-0.082938	-0.189583	-0.320419	-0.327091	-0.179829	-0.168544	
divorced	0.050734	0.019299	-0.166136	-0.379534	-0.398079	-0.361172	-0.409739	-0.342244	

In [39]:
```
plt.figure(figsize=(25,20))
sns.heatmap(train_correl.corr(),annot=True,cmap='coolwarm')
plt.show()
```


COUNTYID 1 0.22 -0.00057 -0.1 -0.08 -0.077 -0.14 -0.092 -0.03 -0.04 -0.13 -0.087 -0.065 -0.058 -0.005 -0.022 0.069 0.051
STATEID 0.22 1 -0.032 -0.22 -0.086 -0.072 -0.17 -0.014 -0.1 -0.11 -0.15 -0.16 -0.025 -0.018 0.071 0.027 0.03 0.019
pop -0.00057 -0.032 1 0.16 0.17 0.13 0.12 0.057 0.085 0.085 0.11 0.25 -0.2 -0.2 0.095 0.17 -0.083 -0.17
rent_mean -0.1 -0.22 0.16 1 0.76 0.7 0.75 0.6 0.13 0.15 0.41 0.44 0.044 0.0035 0.14 0.26 -0.19 -0.38
hi_mean -0.08 -0.086 0.17 0.76 1 0.96 0.77 0.68 0.083 0.1 0.48 0.42 0.2 0.13 0.48 0.54 -0.32 -0.4
family_mean -0.077 -0.072 0.13 0.7 0.96 1 0.76 0.69 0.058 0.077 0.47 0.38 0.26 0.2 0.45 0.49 -0.33 -0.36
hc_mortgage_mean -0.14 -0.17 0.12 0.75 0.77 0.76 1 0.8 0.11 0.15 0.47 0.39 0.096 0.051 0.065 0.23 -0.18 -0.41
hc_mean -0.092 -0.014 0.057 0.6 0.68 0.69 0.8 1 0.033 0.055 0.36 0.3 0.12 0.091 0.098 0.2 -0.17 -0.34
home_equity_second_mortgage -0.03 -0.1 0.085 0.13 0.083 0.058 0.11 0.033 1 0.93 0.52 0.33 -0.14 -0.14 -0.054 -0.0097 -0.0087 -0.042
second_mortgage -0.04 -0.11 0.085 0.15 0.1 0.077 0.15 0.055 0.93 1 0.51 0.36 -0.14 -0.14 -0.055 -0.011 -0.011 -0.059
home_equity -0.13 -0.15 0.11 0.41 0.48 0.47 0.47 0.36 0.52 0.51 1 0.53 0.023 0.011 0.14 0.19 -0.16 -0.21
debt -0.087 -0.16 0.25 0.44 0.42 0.38 0.39 0.3 0.33 0.36 0.53 1 -0.26 -0.27 0.028 0.11 -0.12 -0.23
male_age_mean -0.065 -0.025 -0.2 0.044 0.2 0.26 0.096 0.12 -0.14 -0.14 0.023 -0.26 1 0.86 0.46 0.41 -0.11 0.2
female_age_mean -0.058 -0.018 -0.2 0.0035 0.13 0.2 0.051 0.091 -0.14 -0.14 0.011 -0.27 0.86 1 0.42 0.33 -0.083 0.18
pct_own -0.005 0.071 0.095 0.14 0.48 0.45 0.065 0.098 -0.054 -0.055 0.14 0.028 0.46 0.42 1 0.69 -0.29 -0.1
married -0.022 0.027 0.17 0.26 0.54 0.49 0.23 0.2 -0.0097 -0.011 0.19 0.11 0.41 0.33 0.69 1 -0.22 -0.28
separated 0.069 0.03 -0.083 -0.19 -0.32 -0.33 -0.18 -0.17 -0.0087 -0.011 -0.16 -0.12 -0.11 -0.083 -0.29 -0.22 1 0.13
divorced 0.051 0.019 -0.17 -0.38 -0.4 -0.36 -0.41 -0.34 -0.042 -0.059 -0.21 -0.23 0.2 0.18 -0.1 -0.28 0.13 1
1.0 0.8 0.6 0.4 0.2 0.0 −0.2 −0.4

Computing "Bad Debt" column

In [40]:
```
train_data['Bad_Debt'] = train_data['second_mortgage'] + train_data['home_equity'] - train_data['home_equity_second_mortgage']
```

define category columns

In [41]:
```
category_columns = ['COUNTYID', 'STATEID', 'state', 'state_ab', 'city', 'place', 'type', 'zip_code', 'area_code']
```

define unwanted columns

In [42]:
```
unwanted_columns = ['lat', 'lng','ALand','AWater']
```

Remove category and unwanted columns

In [43]:
```
fa_train_data = train_data.drop(category_columns, axis=1)
fa_train_data = fa_train_data.drop(unwanted_columns, axis=1)
```

In [44]:
```
fa_train_data
```

Out[44]:

UID	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_
267822	5230	2612	2618	769.38638	784.0	232.63967	272.34441	362.0	0.86761	0.79155	0.59155	0.
246444	2633	1349	1284	804.87924	848.0	253.46747	312.58622	513.0	0.97410	0.93227	0.69920	0.
245683	6881	3643	3238	742.77365	703.0	323.39011	291.85520	378.0	0.95238	0.88624	0.79630	0.
279653	2700	1141	1559	803.42018	782.0	297.39258	259.30316	368.0	0.94693	0.87151	0.69832	0.
247218	5637	2586	3051	938.56493	881.0	392.44096	1005.42886	1704.0	0.99286	0.98247	0.91688	0.
...	...	...	...	...	...	...	...	...	...	...	...	
279212	1847	909	938	439.42839	419.0	140.29970	170.00000	170.0	1.00000	1.00000	1.00000	0.
277856	4155	2116	2039	1813.19253	1788.0	492.92300	64.84927	471.0	0.85435	0.63261	0.50000	0.
233000	2829	1465	1364	849.39107	834.0	336.47530	120.91448	195.0	0.93846	0.71282	0.54359	0.
287425	11542	5727	5815	1972.45746	1843.0	633.02173	19.16328	157.0	1.00000	1.00000	0.75796	0.
265371	3726	1815	1911	949.84199	924.0	198.82109	555.87526	1031.0	0.94956	0.87779	0.83705	0.

26954 rows × 64 columns

In [45]:
```
fa_train_data.corr()
```

Out[45]:

	pop	male_pop	female_pop	rent_mean	rent_median	rent_stdev	rent_sample_weight	rent_samples	rent_gt_10	ren
pop	1.000000	0.982546	0.983029	0.163491	0.157255	0.118588	0.247947	0.407705	0.064774	(
male_pop	0.982546	1.000000	0.931747	0.160866	0.155413	0.109425	0.225631	0.385110	0.051229	(
female_pop	0.983029	0.931747	1.000000	0.160492	0.153696	0.123571	0.261480	0.416051	0.075918	(
rent_mean	0.163491	0.160866	0.160492	1.000000	0.976533	0.655728	-0.390880	-0.016583	0.101420	(
rent_median	0.157255	0.155413	0.153696	0.976533	1.000000	0.569753	-0.385541	-0.021226	0.099590	(
rent_stdev	0.118588	0.109425	0.123571	0.655728	0.569753	1.000000	-0.180248	0.066453	-0.014252	(
rent_sample_weight	0.247947	0.225631	0.261480	-0.390880	-0.385541	-0.180248	1.000000	0.802122	0.054375	(
rent_samples	0.407705	0.385110	0.416051	-0.016583	-0.021226	0.066453	0.802122	1.000000	0.104053	(
rent_gt_10	0.064774	0.051229	0.075918	0.101420	0.099590	-0.014252	0.054375	0.104053	1.000000	(
rent_gt_15	0.060412	0.045213	0.073335	0.113683	0.114534	0.040250	0.090222	0.146568	0.616890	1
rent_gt_20	0.021028	0.006773	0.034367	0.054679	0.057573	0.040463	0.126019	0.157692	0.453919	(
rent_gt_25	-0.002041	-0.015432	0.011234	0.011222	0.013398	0.034539	0.141565	0.150293	0.368279	(
rent_gt_30	-0.014094	-0.026944	-0.000940	-0.003690	-0.002387	0.021017	0.122921	0.128968	0.313173	(
rent_gt_35	-0.022559	-0.033761	-0.010741	0.002139	0.002529	0.024627	0.104302	0.111491	0.280114	(
rent_gt_40	-0.030028	-0.040805	-0.018374	-0.002766	-0.003106	0.027704	0.098838	0.102533	0.256114	(
rent_gt_50	-0.046293	-0.056349	-0.034793	-0.004439	-0.006792	0.041014	0.090839	0.094931	0.217357	(
universe_samples	0.423674	0.400898	0.431653	-0.038977	-0.042594	0.052617	0.805515	0.995021	0.097223	(
used_samples	0.413499	0.390557	0.421990	-0.011555	-0.016328	0.068196	0.796272	0.997736	0.103787	(
hi_mean	0.174299	0.176470	0.166200	0.755850	0.719598	0.549125	-0.477675	-0.222931	-0.099060	-(
hi_median	0.180786	0.183688	0.171743	0.752719	0.722732	0.514621	-0.487645	-0.246623	-0.094156	-(
hi_stdev	0.131716	0.128995	0.129898	0.646592	0.600435	0.577344	-0.370783	-0.121765	-0.103705	-(
hi_sample_weight	0.712059	0.676161	0.723120	-0.277390	-0.270732	-0.162372	0.664101	0.637266	0.089542	(
hi_samples	0.899576	0.861267	0.906606	0.105894	0.094173	0.118076	0.371462	0.527922	0.051060	(
family_mean	0.134765	0.134387	0.130532	0.702992	0.660208	0.560268	-0.424119	-0.175046	-0.100577	-(

	pop	male_pop	female_pop	rent_mean	rent_median	rent_stdev	rent_sample_weight	rent_samples	rent_gt_10	ren
family_median	0.130585	0.130167	0.126534	0.699953	0.660504	0.547772	-0.422949	-0.186724	-0.105973	-(
family_stdev	0.112231	0.109186	0.111397	0.566350	0.521023	0.529584	-0.293820	-0.057428	-0.076446	-(
family_sample_weight	0.786870	0.760292	0.786184	-0.218621	-0.201386	-0.197172	0.398699	0.375061	0.078737	(
family_samples	0.932933	0.902190	0.931361	0.165194	0.159147	0.099172	0.109201	0.233331	0.035023	(
hc_mortgage_mean	0.115557	0.113345	0.113788	0.753801	0.715579	0.639646	-0.265316	0.064039	0.011145	(
hc_mortgage_median	0.111318	0.109194	0.109607	0.752775	0.716390	0.633986	-0.256702	0.066092	0.010098	(
hc_mortgage_stdev	0.085424	0.084404	0.083511	0.568042	0.528982	0.528025	-0.263760	0.000941	-0.028454	-(
hc_mortgage_sample_weight	0.646117	0.622569	0.647254	-0.151363	-0.142126	-0.172385	0.039384	-0.043443	0.005186	-(
hc_mortgage_samples	0.775982	0.750081	0.775001	0.290865	0.277407	0.177524	-0.154408	-0.038574	0.031813	-(
hc_mean	0.056816	0.047831	0.063734	0.596587	0.563763	0.510206	-0.208941	0.033162	-0.010495	-(
hc_median	0.055508	0.046097	0.062892	0.578904	0.548866	0.489452	-0.194591	0.039137	-0.008837	(
hc_stdev	0.055423	0.052040	0.056865	0.449156	0.417672	0.442264	-0.181162	0.026044	-0.042442	-(
hc_samples	0.460030	0.443139	0.460963	-0.144777	-0.147252	-0.081819	-0.068086	-0.157668	-0.095781	-(
hc_sample_weight	0.388110	0.376694	0.386101	-0.299954	-0.290918	-0.217486	-0.004893	-0.156283	-0.096360	-(
home_equity_second_mortgage	0.085189	0.081921	0.085500	0.129229	0.129117	0.060677	0.002893	0.081580	0.072426	(
second_mortgage	0.085077	0.081554	0.085644	0.153189	0.151883	0.081208	-0.007712	0.081063	0.078583	(
home_equity	0.105959	0.101359	0.106873	0.413234	0.391111	0.308109	-0.157572	-0.005411	0.070552	(
debt	0.248503	0.237351	0.251006	0.438072	0.425478	0.273010	-0.074634	0.132094	0.149856	(
second_mortgage_cdf	-0.153255	-0.147392	-0.153799	-0.185748	-0.180282	-0.105860	0.109393	0.039256	-0.062519	-(
home_equity_cdf	-0.132398	-0.124193	-0.135965	-0.428718	-0.405314	-0.318689	0.183689	0.029270	-0.074894	-(
debt_cdf	-0.254815	-0.244452	-0.256324	-0.459564	-0.446482	-0.283767	0.082925	-0.138446	-0.149336	-(
hs_degree	0.046917	0.031908	0.060115	0.364746	0.334504	0.271358	-0.257882	-0.118794	-0.051549	-(
hs_degree_male	0.055718	0.039181	0.070121	0.373519	0.343103	0.284382	-0.237279	-0.092346	-0.044940	-(
hs_degree_female	0.037123	0.029685	0.043190	0.330269	0.302126	0.239690	-0.264723	-0.141198	-0.057185	-(

	pop	male_pop	female_pop	rent_mean	rent_median	rent_stdev	rent_sample_weight	rent_samples	rent_gt_10	ren
male_age_mean	-0.195106	-0.208192	-0.175532	0.043815	0.024789	0.110554	-0.271333	-0.298355	-0.097756	-(
male_age_median	-0.157367	-0.167835	-0.141664	0.093181	0.075080	0.116560	-0.343486	-0.363057	-0.105940	-(
male_age_stdev	-0.040834	-0.082516	0.001669	-0.116530	-0.119319	-0.013115	-0.138071	-0.278975	-0.062663	-(
male_age_sample_weight	0.917564	0.944742	0.859397	0.133127	0.126641	0.097127	0.239284	0.376077	0.056553	(
male_age_samples	0.982227	0.999702	0.931418	0.160751	0.155334	0.109099	0.225457	0.384982	0.050204	(
female_age_mean	-0.200166	-0.203842	-0.189698	0.003500	-0.014405	0.102461	-0.208829	-0.255675	-0.070820	-(
female_age_median	-0.166636	-0.164599	-0.162949	0.045057	0.028790	0.099439	-0.304827	-0.351972	-0.083306	-(
female_age_stdev	-0.038493	-0.067619	-0.008454	-0.180747	-0.183773	-0.027092	-0.035609	-0.186171	-0.050657	-(
female_age_sample_weight	0.925712	0.876698	0.942404	0.129438	0.121430	0.108656	0.287806	0.417133	0.080870	(
female_age_samples	0.982833	0.931673	0.999689	0.160537	0.153764	0.123204	0.261312	0.415896	0.076089	(
pct_own	0.095063	0.093931	0.092931	0.140578	0.132140	0.049538	-0.613282	-0.686398	-0.098457	-(
married	0.167639	0.140539	0.188634	0.258521	0.245020	0.131570	-0.445051	-0.396995	-0.094542	-(
married_snp	-0.034160	-0.005206	-0.061546	-0.109462	-0.097351	-0.071324	0.220034	0.216787	0.037976	(
separated	-0.082938	-0.077443	-0.085522	-0.189583	-0.175533	-0.138012	0.206038	0.142461	0.042156	(
divorced	-0.166136	-0.153991	-0.172435	-0.379534	-0.363223	-0.272010	0.214523	0.052899	-0.010217	-(
Bad Debt	0.106022	0.101214	0.107042	0.416821	0.394822	0.210612	0.157522	0.092191	0.074178	(

Performing Kaiser-Meyer-Olkin (KMO) Test to measure the suitability of data for factor analysis

In [46]:
```
from factor_analyzer import FactorAnalyzer
```

In [47]:
```
from factor_analyzer.factor_analyzer import calculate_kmo
```

In [48]:
```
kmo_all,kmo_model=calculate_kmo(fa_train_data)
```

```
C:\Users\14sac\anaconda3\lib\site-packages\factor_analyzer\utils.py:244: UserWarning: The inverse of the variance-covariance mat
rix was calculated using the Moore-Penrose generalized matrix inversion, due to its determinant being at or very close to zero.
  warnings.warn(
```

In [49]:
```
kmo_model
```

Out[49]:
```
0.8690059170820098
```

In [50]:
```
fa = FactorAnalyzer(rotation=None, n_factors = 25)
fa.fit(fa_train_data)
```

Out[50]:
```
FactorAnalyzer(n_factors=25, rotation=None, rotation_kwargs={})
```

In [51]:
```
ev, v = fa.get_eigenvalues()
sorted(ev,reverse=True)
```

Out[51]:
```
[15.480602432103586,
 12.035298223423114,
 8.165675040227022,
 4.560327457433876,
 3.938267006278989,
 3.0385535659292953,
 2.1328597186434965,
 1.4376210328626557,
 1.314363398590606,
 1.1417681229735344,
 0.9759649736686924,
 0.939478391319101,
 0.8316062246346417,
 0.7524797022206464,
 0.6558022438811698,
 0.5706593917411812,
 0.5274510677941903,
 0.4835517718964641,
 0.4569310116319864,
 0.3917697613974229,
 0.3651669458039393,
 0.31995971111724025,
 0.3136128003737653,
 0.3044376689561306,
 0.2588531647522717,
 0.24364519533519738,
 0.2282959535626511,
 0.20273674543147266,
 0.19800697363097755,
 0.1892872507903567,
 0.16656170180881577,
 0.15928452393563222,
 0.1328301478012298,
 0.1318312431223947,
 0.11850174363284335,
 0.10847018757996843,
 0.09965038105500786,
 0.09292330252129448,
 0.08975978881244928,
 0.0577491562681048 2,
 0.05548316216407655,
 0.04705552031873865,
 0.04598166786464995,
 0.03498789556849009,
```

```
0.03259132010805165,
0.02709757172147369,
0.024287701804966522,
0.02008706429086253,
0.016455809191409893,
0.015586895752882786,
0.014727844985200833,
0.0138882373162985,
0.009562701245507435,
0.008484411198401655,
0.006895718660330041,
0.003960213082553237,
0.003579697813549775,
0.0025637262734029337,
0.0019259341476483057,
0.0015997775499564002,
0.00031895150764140297,
0.00028305249049798815,
-1.2993652042300832e-17,
-7.091844867439415e-16]
```

In [52]:
```
loadings = fa.loadings_
pd.DataFrame(fa.loadings_,index=fa_train_data.columns)
```

Out[52]:

	0	1	2	3	4	5	6	7	8	9	10	
pop	0.313283	0.929977	-0.059797	0.033344	0.022319	-0.113993	0.059267	-0.000596	0.014089	0.013495	-0.004183	0.(
male_pop	0.303881	0.905419	-0.055115	0.011135	0.024456	-0.140267	0.070857	-0.024957	0.059980	0.029983	-0.026433	0.(
female_pop	0.311753	0.922055	-0.062335	0.054051	0.019426	-0.084055	0.045709	0.023367	-0.031524	-0.003210	0.017811	-0.(
rent_mean	0.742346	-0.076398	0.358525	0.148899	0.097031	-0.174349	0.030844	0.065852	0.226355	-0.080250	0.132828	-0.2
rent_median	0.700926	-0.070701	0.349216	0.134462	0.078146	-0.187808	0.034698	0.072315	0.231236	-0.086420	0.130243	-0.2
rent_stdev	0.561891	-0.060961	0.264913	0.159563	0.204521	0.008993	0.072023	-0.010279	0.014692	0.010126	0.058088	-0.1
rent_sample_weight	-0.409278	0.490876	0.231803	-0.185377	0.267474	0.449327	-0.056424	0.040995	-0.140220	0.014037	-0.009360	0.(
rent_samples	-0.127088	0.574391	0.458955	-0.183258	0.391686	0.469318	-0.062596	0.044793	-0.021532	-0.015490	0.045727	-0.(
rent_gt_10	-0.061813	0.116527	0.265555	0.296110	-0.167290	0.028954	-0.137631	0.225287	0.116928	-0.273955	0.005822	0.1
rent_gt_15	-0.124908	0.137598	0.417465	0.503291	-0.189056	0.032982	-0.173814	0.299458	0.167426	-0.417475	0.018670	0.2
rent_gt_20	-0.222559	0.120862	0.456485	0.599105	-0.146487	0.027169	-0.124081	0.150827	0.072229	-0.206982	0.010568	0.1
rent_gt_25	-0.286348	0.110881	0.475960	0.682705	-0.124533	0.003276	-0.105184	0.037082	-0.002758	-0.048683	0.009459	0.(
rent_gt_30	-0.309183	0.099414	0.459158	0.700347	-0.110492	-0.032820	-0.095403	-0.076236	-0.057857	0.101654	0.023070	-0.(
rent_gt_35	-0.304729	0.083853	0.447520	0.705896	-0.090390	-0.060191	-0.099125	-0.150421	-0.097400	0.201819	0.025361	-0.(
rent_gt_40	-0.301994	0.073096	0.437014	0.693665	-0.080359	-0.069299	-0.103862	-0.203034	-0.124957	0.264893	0.022615	-0.1
rent_gt_50	-0.278294	0.047764	0.398967	0.597269	-0.049552	-0.062377	-0.088922	-0.196010	-0.116134	0.229780	0.012628	-0.1
universe_samples	-0.144491	0.594334	0.429733	-0.174175	0.398748	0.466105	-0.052571	0.034155	-0.026930	-0.017386	0.049316	-0.(
used_samples	-0.118790	0.577391	0.453980	-0.184315	0.388859	0.472065	-0.058022	0.053269	-0.025968	-0.015503	0.051846	-0.(
hi_mean	0.942924	-0.148376	0.016519	-0.012098	0.074172	-0.160193	0.017880	0.024712	-0.006447	0.021145	0.066876	-0.(
hi_median	0.913922	-0.135566	0.004441	-0.040415	0.022411	-0.203543	0.014634	0.049932	0.017800	0.010291	0.066242	-0.(
hi_stdev	0.876035	-0.158395	0.044641	0.080268	0.211237	-0.013834	0.023822	-0.054075	-0.098834	0.060446	0.070979	0.(
hi_sample_weight	-0.179975	0.875568	-0.158313	0.068002	0.114039	0.324363	-0.036997	0.011817	-0.023878	-0.014019	-0.011594	-0.(
hi_samples	0.326721	0.878775	-0.165011	0.047416	0.113826	0.196187	-0.049163	0.025870	-0.002764	0.010446	0.019784	-0.(
family_mean	0.935045	-0.173394	0.007553	0.003258	0.134458	-0.058131	-0.051533	-0.021503	-0.021237	0.034895	0.053966	0.(

	0	1	2	3	4	5	6	7	8	9	10	
family_median	0.910837	-0.172002	0.005615	-0.006585	0.118357	-0.089167	-0.045677	-0.012355	-0.017028	0.029105	0.055787	-0.(
family_stdev	0.807378	-0.142039	0.054232	0.074517	0.224817	0.073153	-0.033977	-0.073467	-0.093061	0.068202	0.065927	0.(
family_sample_weight	-0.138478	0.879374	-0.228153	0.098211	-0.047519	-0.005233	0.130578	0.057020	-0.030997	-0.041045	-0.008318	-0.(
family_samples	0.402639	0.834980	-0.255641	0.089832	-0.052020	-0.090478	0.072239	0.063466	-0.029410	-0.001818	0.017041	-0.(
hc_mortgage_mean	0.782582	-0.134456	0.399172	0.110813	0.278373	-0.012660	0.181708	-0.036542	-0.000606	-0.062357	-0.046178	-0.(
hc_mortgage_median	0.762905	-0.132667	0.407617	0.103598	0.272322	-0.027852	0.177390	-0.025553	-0.002688	-0.059963	-0.039509	-0.(
hc_mortgage_stdev	0.682338	-0.137306	0.176456	0.144028	0.251454	0.082347	0.167524	-0.086007	-0.015421	-0.033453	-0.039148	0.(
hc_mortgage_sample_weight	0.125808	0.643071	-0.508041	0.030002	-0.312282	-0.082073	-0.142382	0.089487	0.026939	0.083411	-0.012889	-0.(
hc_mortgage_samples	0.593631	0.604933	-0.277840	0.070927	-0.244426	-0.152228	-0.077421	0.108705	0.038087	0.074078	-0.017837	0.(
hc_mean	0.700692	-0.167291	0.306114	0.109565	0.340584	0.012891	0.099005	-0.039237	-0.121820	-0.073289	-0.073969	0.2
hc_median	0.670234	-0.157863	0.306112	0.099226	0.326220	0.005583	0.089838	-0.032836	-0.117497	-0.074190	-0.069839	0.1
hc_stdev	0.539030	-0.127939	0.146857	0.144208	0.347649	0.047659	0.130308	-0.090912	-0.067675	-0.042308	-0.005709	0.1
hc_samples	0.118346	0.415698	-0.717215	0.286913	0.078912	-0.018958	0.059552	-0.254567	0.057757	-0.076072	-0.048362	0.1
hc_sample_weight	-0.091109	0.413973	-0.741961	0.236656	0.011960	-0.031680	0.053050	-0.245707	0.077906	-0.061793	-0.030183	0.(
home_equity_second_mortgage	0.181501	0.089349	0.295222	-0.154513	-0.597113	0.306578	0.279866	-0.292792	0.090365	-0.111107	0.265598	-0.(
second_mortgage	0.203466	0.083554	0.321341	-0.146359	-0.617397	0.318727	0.308346	-0.320388	0.102287	-0.129314	0.308434	-0.(
home_equity	0.595258	-0.024765	0.300193	-0.062733	-0.475440	0.304826	0.136570	-0.035713	-0.093163	0.067151	-0.359837	0.1
debt	0.497193	0.159865	0.445275	-0.190914	-0.378737	-0.018960	-0.128144	0.373529	0.010823	0.148459	-0.042232	-0.1
second_mortgage_cdf	-0.329114	-0.096298	-0.128230	0.074017	0.573851	-0.234200	-0.178399	0.135317	-0.006817	0.052716	-0.211122	0.(
home_equity_cdf	-0.638701	0.013781	-0.247682	0.036563	0.467459	-0.295244	-0.092205	-0.017819	0.116134	-0.072082	0.286845	-0.(
debt_cdf	-0.493580	-0.162470	-0.482873	0.205111	0.372691	0.057949	0.120894	-0.372765	-0.042352	-0.150008	0.053492	0.1
hs_degree	0.673174	-0.122592	-0.196543	-0.065730	-0.066774	0.259380	-0.570642	-0.061101	0.040862	0.113066	0.150697	0.1
hs_degree_male	0.664874	-0.111205	-0.163247	-0.065865	-0.050610	0.255448	-0.548787	-0.051509	0.030692	0.104853	0.148799	0.1
hs_degree_female	0.637788	-0.126876	-0.220088	-0.063518	-0.076621	0.244497	-0.550218	-0.069713	0.059563	0.120166	0.138420	0.1

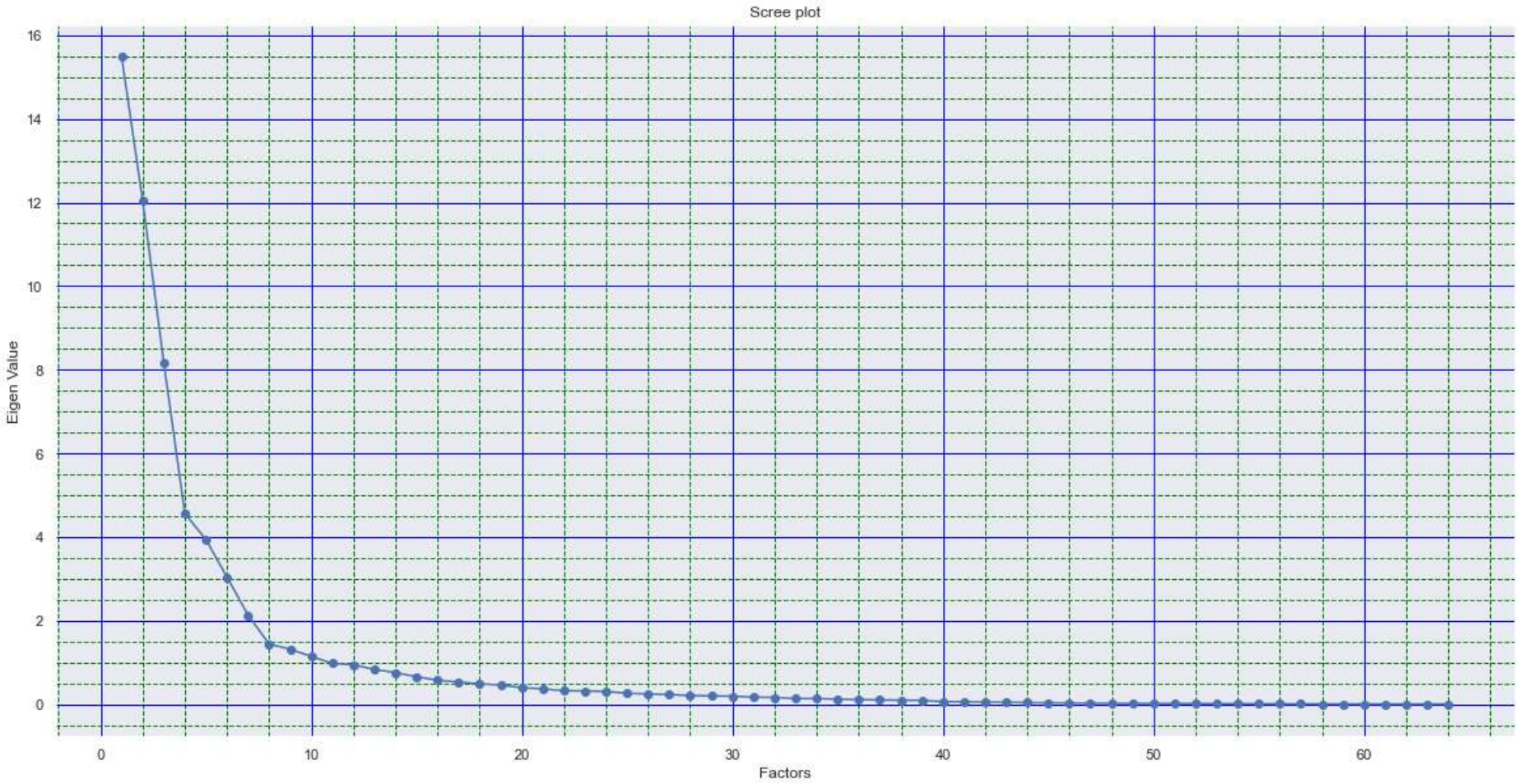
	0	1	2	3	4	5	6	7	8	9	10	
male_age_mean	0.250202	-0.310447	-0.564585	0.379921	0.125052	0.416835	0.075607	0.027940	0.226165	0.019142	-0.094845	-0.
male_age_median	0.324365	-0.305576	-0.577257	0.368416	0.069827	0.343492	0.090407	0.042053	0.250583	0.033885	-0.109089	-0.
male_age_stdev	0.073840	-0.098934	-0.499386	0.296809	-0.066488	0.177551	0.236310	0.279518	-0.462784	-0.020920	0.230888	0.(
male_age_sample_weight	0.239715	0.858425	0.014986	-0.005617	0.036568	-0.183386	-0.011188	-0.135778	0.089872	0.022303	-0.075691	0.(
male_age_samples	0.303862	0.905276	-0.055305	0.010718	0.024428	-0.140338	0.070526	-0.025377	0.061198	0.030221	-0.027191	0.(
female_age_mean	0.189210	-0.292351	-0.546557	0.415727	0.116754	0.476687	0.083943	0.081364	0.166014	0.008204	-0.078054	-0.
female_age_median	0.260919	-0.292986	-0.578939	0.406551	0.049312	0.375176	0.096943	0.078766	0.171876	0.014311	-0.086879	-0.
female_age_stdev	-0.019351	-0.066170	-0.426865	0.240459	-0.037080	0.210502	0.249304	0.293105	-0.462506	0.018496	0.221205	0.(
female_age_sample_weight	0.246210	0.886961	0.016142	0.028425	0.034910	-0.131238	-0.037905	-0.097900	-0.000470	-0.008399	-0.034522	0.(
female_age_samples	0.311600	0.921996	-0.062283	0.053873	0.019328	-0.084186	0.045186	0.022981	-0.030166	-0.003397	0.017206	-0.(
pct_own	0.473327	-0.129383	-0.636793	0.220977	-0.300618	-0.229041	0.005852	0.047555	-0.061384	0.020562	0.013413	0.(
married	0.546064	-0.039487	-0.492664	0.130780	-0.130295	-0.113319	0.140881	0.184225	-0.071015	0.002900	0.115265	-0.(
married_snp	-0.355507	0.055497	0.282245	0.010724	0.168485	0.000919	0.481629	0.213119	0.377151	0.435162	0.154432	0.:
separated	-0.354169	0.026608	0.146540	0.023213	0.074763	0.058836	0.284467	0.171355	0.223334	0.297633	0.118243	0.:
divorced	-0.379574	-0.033229	-0.208450	0.012799	-0.039098	0.319233	-0.087352	0.046277	0.065064	0.041523	0.023158	-0.(
Bad Debt	0.592682	0.022472	0.212098	0.060921	0.482617	0.205171	0.147861	0.046428	0.082286	0.056671	0.224226	0.(

In [53]:
```
xvals=range(1,fa_train_data.shape[1]+1)
xvals
```

Out[53]:
```
range(1, 65)
```

In [54]:
```
sns.set()
plt.figure(figsize=(20,10))
plt.scatter(xvals,ev)
plt.plot(xvals, ev)
plt.title('Scree plot')
plt.xlabel('Factors')
plt.ylabel('Eigen Value')
```

```
plt.grid(color = 'blue', )
plt.grid(visible=True, which='minor', color='green', linestyle='--')
plt.minorticks_on()
plt.show()
```



In [55]:
```
# selected 10 factors as the optimum number of factors, since their Eigenvalues is greater than 1.
```

In [56]:
```
fa=FactorAnalyzer(rotation='varimax',n_factors=10)
fact = fa.fit(fa_train_data)
loadings=fa.loadings_
```

In [57]:
```
Factors = pd.DataFrame(loadings,index=fa_train_data.columns)
Factors
```

Out[57]:

	0	1	2	3	4	5	6	7	8	9
pop	0.120128	0.973605	-0.000183	0.113105	-0.102589	-0.010924	0.035808	0.021067	-0.015870	0.034228
male_pop	0.117765	0.948401	-0.018944	0.091822	-0.104819	-0.032709	0.032793	0.023786	-0.070903	0.022289
female_pop	0.117420	0.959158	0.018591	0.131416	-0.096366	0.013653	0.037945	0.017681	0.040780	0.045076
rent_mean	0.803981	0.071197	0.083852	-0.119779	-0.022351	0.070690	0.104968	0.003840	-0.152613	0.264957
rent_median	0.758641	0.069522	0.078893	-0.128204	-0.035387	0.054991	0.105381	0.000347	-0.154861	0.268750
rent_stdev	0.668924	0.033145	0.081153	0.045692	0.041983	0.047313	0.038305	0.035312	-0.008464	0.047418
rent_sample_weight	-0.313168	0.200895	0.048689	0.787903	-0.117392	-0.098206	-0.042510	0.015689	0.049225	-0.056956
rent_samples	0.024695	0.292222	0.075993	0.919368	-0.143434	-0.082800	0.016737	-0.007209	-0.078774	0.058254
rent_gt_10	-0.029198	0.055869	0.406585	0.044740	0.007173	-0.012542	0.052354	0.011869	-0.034423	0.333288
rent_gt_15	-0.015221	0.043261	0.629354	0.074849	0.008398	-0.062044	0.054809	-0.005076	-0.043489	0.373853
rent_gt_20	-0.039987	0.006921	0.779502	0.090239	0.004230	-0.112928	0.034149	-0.014509	-0.025177	0.256254
rent_gt_25	-0.055967	-0.011760	0.866930	0.080656	-0.022494	-0.131013	0.011578	-0.007998	-0.008509	0.104578
rent_gt_30	-0.062039	-0.017779	0.909530	0.049611	-0.056055	-0.114425	0.000505	-0.006053	-0.002811	-0.036870
rent_gt_35	-0.048451	-0.025362	0.911176	0.029172	-0.081709	-0.092653	-0.018171	0.000793	-0.003832	-0.133493
rent_gt_40	-0.046689	-0.033267	0.878634	0.025236	-0.095007	-0.080528	-0.023866	0.006727	-0.009572	-0.179004
rent_gt_50	-0.036425	-0.052069	0.776030	0.034536	-0.098412	-0.076608	-0.029449	0.003673	-0.017001	-0.185438
universe_samples	0.004340	0.314471	0.073696	0.915304	-0.132378	-0.091209	0.010361	-0.020375	-0.067570	0.035069
used_samples	0.028779	0.297508	0.070534	0.918102	-0.141261	-0.081660	0.020201	-0.006085	-0.067288	0.064565
hi_mean	0.841310	0.108502	-0.227804	-0.271702	0.028803	0.281443	0.056958	0.093754	0.030069	0.109371
hi_median	0.788932	0.120941	-0.242617	-0.312645	0.001758	0.267424	0.065872	0.093727	0.012933	0.152614
hi_stdev	0.846835	0.058658	-0.139281	-0.110268	0.098541	0.280883	0.020121	0.082274	0.090357	-0.041971
hi_sample_weight	-0.325871	0.753260	0.079915	0.485866	0.133638	-0.006263	-0.011081	-0.024732	0.077794	-0.056456
hi_samples	0.089498	0.899097	-0.043841	0.317094	0.120953	0.157132	0.026462	0.024321	0.063920	0.026639
family_mean	0.836808	0.067490	-0.212623	-0.183494	0.085536	0.359741	0.033074	0.096448	0.021609	0.049376

	0	1	2	3	4	5	6	7	8	9
family_median	0.811073	0.067367	-0.215013	-0.205104	0.065140	0.340055	0.031102	0.089090	0.018193	0.062404
family_stdev	0.769602	0.040279	-0.116033	-0.025721	0.122203	0.311940	0.019178	0.086245	0.066113	-0.049414
family_sample_weight	-0.308839	0.844558	0.059050	0.138999	0.020980	-0.165477	0.011854	-0.039144	0.117249	-0.009778
family_samples	0.122173	0.952737	-0.064883	-0.031770	0.040341	0.067826	0.043086	0.033941	0.120283	0.061167
hc_mortgage_mean	0.941529	-0.006157	0.016282	0.050841	0.005740	-0.011379	0.062525	0.127510	-0.056877	0.032463
hc_mortgage_median	0.925465	-0.010292	0.020184	0.048272	-0.014518	-0.018392	0.057923	0.121161	-0.055949	0.044467
hc_mortgage_stdev	0.766753	0.006880	-0.027162	0.023713	0.155766	0.047904	0.053126	0.108400	0.009177	-0.069240
hc_mortgage_sample_weight	-0.307726	0.754275	-0.108229	-0.220573	0.113125	0.247736	0.020781	0.052151	0.090653	0.107536
hc_mortgage_samples	0.204322	0.795419	-0.097237	-0.253994	0.056764	0.275779	0.079621	0.139626	0.051145	0.196629
hc_mean	0.836964	-0.047494	-0.000646	0.071362	0.013299	0.076515	-0.024279	0.084451	0.016355	-0.028317
hc_median	0.802620	-0.046288	0.001655	0.071392	-0.000304	0.072881	-0.023935	0.080053	0.011311	-0.018306
hc_stdev	0.676242	-0.015974	-0.010937	0.069210	0.115829	0.030510	-0.032377	-0.003632	0.040855	-0.124205
hc_samples	-0.124229	0.607940	-0.091780	-0.235451	0.462251	0.113352	-0.094024	-0.152015	0.085783	-0.363649
hc_sample_weight	-0.331545	0.556749	-0.089245	-0.233250	0.418113	0.058523	-0.089877	-0.177463	0.075603	-0.350873
home_equity_second_mortgage	0.033825	0.030039	0.021921	0.041897	-0.084969	-0.004138	0.907018	0.177851	-0.053237	0.031426
second_mortgage	0.063359	0.024610	0.034108	0.041042	-0.083456	-0.012849	0.975199	0.154149	-0.054168	0.030842
home_equity	0.371485	0.035098	-0.004853	-0.023904	0.007837	0.148049	0.378432	0.793904	-0.024040	0.081751
debt	0.324560	0.153948	0.010837	0.012861	-0.310623	0.173462	0.226177	0.372772	-0.053069	0.558665
second_mortgage_cdf	-0.097010	-0.120158	0.007563	0.083833	0.018045	-0.126379	-0.704834	-0.225926	-0.073542	-0.114301
home_equity_cdf	-0.389715	-0.065301	0.009030	0.045598	-0.026431	-0.210698	-0.363810	-0.742625	-0.040959	-0.130800
debt_cdf	-0.337608	-0.152717	-0.014136	-0.006361	0.341864	-0.144252	-0.220685	-0.368802	0.093332	-0.578489
hs_degree	0.337273	0.022187	-0.169680	-0.017947	0.196445	0.869586	0.066928	0.074474	-0.036591	0.098295
hs_degree_male	0.353735	0.026584	-0.162551	-0.003487	0.178370	0.803239	0.063636	0.081419	-0.029693	0.102550
hs_degree_female	0.306619	0.022584	-0.177738	-0.048029	0.208407	0.803587	0.062022	0.078222	-0.043300	0.084285

	0	1	2	3	4	5	6	7	8	9
male_age_mean	0.126305	-0.105310	-0.086993	-0.127685	0.896857	0.128950	-0.052772	-0.006577	0.108056	-0.060012
male_age_median	0.166864	-0.067206	-0.112777	-0.219886	0.848955	0.138168	-0.036686	0.015361	0.098812	-0.030040
male_age_stdev	-0.033201	0.028357	-0.040099	-0.179218	0.320689	0.034315	-0.011717	-0.022880	0.832028	-0.045396
male_age_sample_weight	0.087846	0.873666	0.029595	0.097130	-0.170715	0.000903	0.009659	0.017772	-0.190202	-0.047719
male_age_samples	0.117530	0.948327	-0.019356	0.091643	-0.104395	-0.032631	0.032716	0.023881	-0.072281	0.022381
female_age_mean	0.078571	-0.114698	-0.038084	-0.074901	0.875488	0.115904	-0.052924	0.002785	0.196931	-0.045038
female_age_median	0.111078	-0.074549	-0.065371	-0.194958	0.856746	0.124444	-0.038646	0.021943	0.183796	-0.024574
female_age_stdev	-0.092085	0.011067	-0.039245	-0.083379	0.272899	-0.021239	-0.026357	-0.000696	0.746646	-0.050619
female_age_sample_weight	0.087066	0.890766	0.065038	0.143168	-0.172992	0.045741	0.016021	0.013252	-0.093610	-0.033269
female_age_samples	0.117125	0.959116	0.018557	0.131255	-0.095981	0.013749	0.037887	0.017663	0.039163	0.045323
pct_own	0.132285	0.201958	-0.171740	-0.706264	0.322314	0.298721	0.004121	0.046770	0.258775	0.000916
married	0.286290	0.234177	-0.244098	-0.440423	0.292342	0.173919	0.026144	0.026786	0.294599	0.078484
married_snp	-0.087496	-0.074671	0.101849	0.202884	-0.045421	-0.529056	0.002965	-0.065288	-0.076246	0.059467
separated	-0.179021	-0.086037	0.089844	0.163016	0.010694	-0.384570	-0.000374	-0.050681	-0.015090	0.061878
divorced	-0.428368	-0.120273	0.003665	0.168145	0.245704	0.030924	0.001690	-0.049829	0.059020	-0.001477
Bad_Debt	0.375158	0.033209	0.002165	-0.022260	0.003971	0.139839	0.411083	0.773473	-0.025157	0.084829

In [58]:
```
Factor_variance = fa.get_factor_variance()
pd.DataFrame(Factor_variance,index=['SS Loadings', 'Proportion Var', 'Cumulative Var'])
```

Out[58]:

	0	1	2	3	4	5	6	7	8	9
SS Loadings	11.878223	11.630391	5.513000	5.114170	4.525977	3.726219	2.928129	2.384466	1.749791	1.708220
Proportion Var	0.185597	0.181725	0.086141	0.079909	0.070718	0.058222	0.045752	0.037257	0.027340	0.026691
Cumulative Var	0.185597	0.367322	0.453463	0.533372	0.604090	0.662312	0.708064	0.745321	0.772662	0.799353

In [59]:
```
# 80% of cumulative variance is explained by 10 factors
```

In [60]:
```
# Checking the loadings for some of the latent variables
```

In [61]:
```
Factors_df = Factors.loc[['hs_degree','hs_degree_male','hs_degree_female','male_age_median','female_age_median',
                          'home_equity_second_mortgage','second_mortgage','second_mortgage_cdf','pct_own','Bad_Debt']]
```

In [62]:
```
Factors_df
```

Out[62]:

	0	1	2	3	4	5	6	7	8	9
hs_degree	0.337273	0.022187	-0.169680	-0.017947	0.196445	0.869586	0.066928	0.074474	-0.036591	0.098295
hs_degree_male	0.353735	0.026584	-0.162551	-0.003487	0.178370	0.803239	0.063636	0.081419	-0.029693	0.102550
hs_degree_female	0.306619	0.022584	-0.177738	-0.048029	0.208407	0.803587	0.062022	0.078222	-0.043300	0.084285
male_age_median	0.166864	-0.067206	-0.112777	-0.219886	0.848955	0.138168	-0.036686	0.015361	0.098812	-0.030040
female_age_median	0.111078	-0.074549	-0.065371	-0.194958	0.856746	0.124444	-0.038646	0.021943	0.183796	-0.024574
home_equity_second_mortgage	0.033825	0.030039	0.021921	0.041897	-0.084969	-0.004138	0.907018	0.177851	-0.053237	0.031426
second_mortgage	0.063359	0.024610	0.034108	0.041042	-0.083456	-0.012849	0.975199	0.154149	-0.054168	0.030842
second_mortgage_cdf	-0.097010	-0.120158	0.007563	0.083833	0.018045	-0.126379	-0.704834	-0.225926	-0.073542	-0.114301
pct_own	0.132285	0.201958	-0.171740	-0.706264	0.322314	0.298721	0.004121	0.046770	0.258775	0.000916
Bad_Debt	0.375158	0.033209	0.002165	-0.022260	0.003971	0.139839	0.411083	0.773473	-0.025157	0.084829

In [63]:
```
def color(val):
    if val >= 0.5 or val <= -0.5:
        color = 'green'
    else:
        color = 'white'
    return 'background-color: %s' % color
```

In [64]:
```
Factors_df.round(2).style.applymap(color)
```

Out[64]:

	0	1	2	3	4	5	6	7	8	9
hs_degree	0.340000	0.020000	-0.170000	-0.020000	0.200000	0.870000	0.070000	0.070000	-0.040000	0.100000
hs_degree_male	0.350000	0.030000	-0.160000	-0.000000	0.180000	0.800000	0.060000	0.080000	-0.030000	0.100000
hs_degree_female	0.310000	0.020000	-0.180000	-0.050000	0.210000	0.800000	0.060000	0.080000	-0.040000	0.080000
male_age_median	0.170000	-0.070000	-0.110000	-0.220000	0.850000	0.140000	-0.040000	0.020000	0.100000	-0.030000
female_age_median	0.110000	-0.070000	-0.070000	-0.190000	0.860000	0.120000	-0.040000	0.020000	0.180000	-0.020000
home_equity_second_mortgage	0.030000	0.030000	0.020000	0.040000	-0.080000	-0.000000	0.910000	0.180000	-0.050000	0.030000
second_mortgage	0.060000	0.020000	0.030000	0.040000	-0.080000	-0.010000	0.980000	0.150000	-0.050000	0.030000
second_mortgage_cdf	-0.100000	-0.120000	0.010000	0.080000	0.020000	-0.130000	-0.700000	-0.230000	-0.070000	-0.110000
pct_own	0.130000	0.200000	-0.170000	-0.710000	0.320000	0.300000	0.000000	0.050000	0.260000	0.000000
Bad_Debt	0.380000	0.030000	0.000000	-0.020000	0.000000	0.140000	0.410000	0.770000	-0.030000	0.080000

In [65]:
```
# For HS degree, factor 5 explains the maximum variation in these variables
# for Median Age, factor 4 explains the maximum variation in these variables
# for Second mortgage statistics, factor 6 explains the maximum variation in these variables
# for Ownership percentage, factor 3 explains the maximum variation
# for Bad Debt, factor 7 explains the maximum variation
```

Modeling - Linear Regression Model

In [66]:
```
from sklearn.linear_model import LinearRegression
```

In [67]:
```
lm = LinearRegression()
```

Pre-processing Test dataset on similar lines as done for training set

In [68]:
```
test_data = pd.read_csv(r'C:\Sachin new\Simplilearn\Capstone Project\Real Estate Project\test.csv')
```

In [69]:
```
test_data.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 [70]:
```
test_data['UID'].duplicated().sum()
```

Out[70]:
```
32
```

In [71]:
```
test_data.drop_duplicates(inplace = True)
```

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

Out[72]:
```
(11677, 80)
```

In [73]:
```
test_data.set_index('UID',inplace = True)
```

In [74]:
```
test_data.head()
```

Out[74]:

UID	BLOCKID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	type	primary	zip_code	area_code	lat	
255504	NaN	140	163	26	Michigan	MI	Detroit	Dearborn Heights City	CDP	tract	48239	313	42.346422	-8
252676	NaN	140	1	23	Maine	ME	Auburn	Auburn City	City	tract	4210	207	44.100724	-7
276314	NaN	140	15	42	Pennsylvania	PA	Pine City	Millerton	Borough	tract	14871	607	41.948556	-7
248614	NaN	140	231	21	Kentucky	KY	Monticello	Monticello City	City	tract	42633	606	36.746009	-8
286865	NaN	140	355	48	Texas	TX	Corpus Christi	Edroy	Town	tract	78410	361	27.882462	-9

In [75]:
```
pd.DataFrame({'Missing Count': test_data.isna().sum(), 'Missing Percent': (test_data.isna().mean().round(4)*100)})
```

Out[75]:

	Missing Count	Missing Percent
BLOCKID	11677	100.00
SUMLEVEL	0	0.00
COUNTYID	0	0.00
STATEID	0	0.00
state	0	0.00
state_ab	0	0.00
city	0	0.00
place	0	0.00
type	0	0.00
primary	0	0.00
zip_code	0	0.00
area_code	0	0.00
lat	0	0.00
lng	0	0.00
ALand	0	0.00
AWater	0	0.00
pop	0	0.00
male_pop	0	0.00
female_pop	0	0.00
rent_mean	134	1.15
rent_median	134	1.15
rent_stdev	134	1.15
rent_sample_weight	134	1.15
rent_samples	134	1.15

	Missing Count	Missing Percent
rent_gt_10	135	1.16
rent_gt_15	135	1.16
rent_gt_20	135	1.16
rent_gt_25	135	1.16
rent_gt_30	135	1.16
rent_gt_35	135	1.16
rent_gt_40	135	1.16
rent_gt_50	135	1.16
universe_samples	0	0.00
used_samples	0	0.00
hi_mean	112	0.96
hi_median	112	0.96
hi_stdev	112	0.96
hi_sample_weight	112	0.96
hi_samples	112	0.96
family_mean	125	1.07
family_median	125	1.07
family_stdev	125	1.07
family_sample_weight	125	1.07
family_samples	125	1.07
hc_mortgage_mean	245	2.10
hc_mortgage_median	245	2.10
hc_mortgage_stdev	245	2.10
hc_mortgage_sample_weight	245	2.10

	Missing Count	Missing Percent
hc_mortgage_samples	245	2.10
hc_mean	267	2.29
hc_median	267	2.29
hc_stdev	267	2.29
hc_samples	267	2.29
hc_sample_weight	267	2.29
home_equity_second_mortgage	204	1.75
second_mortgage	204	1.75
home_equity	204	1.75
debt	204	1.75
second_mortgage_cdf	204	1.75
home_equity_cdf	204	1.75
debt_cdf	204	1.75
hs_degree	78	0.67
hs_degree_male	82	0.70
hs_degree_female	96	0.82
male_age_mean	77	0.66
male_age_median	77	0.66
male_age_stdev	77	0.66
male_age_sample_weight	77	0.66
male_age_samples	77	0.66
female_age_mean	87	0.75
female_age_median	87	0.75
female_age_stdev	87	0.75

	Missing Count	Missing Percent
female_age_sample_weight	87	0.75
female_age_samples	87	0.75
pct_own	112	0.96
married	77	0.66
married_snp	77	0.66
separated	77	0.66
divorced	77	0.66

In [76]:
```
test_data.drop(test_data[['BLOCKID','primary','SUMLEVEL']], axis = 1, inplace=True)
```

In [77]:
```
(test_data['pop']==0).value_counts()
```

Out[77]:
```
False    11603
True        74
Name: pop, dtype: int64
```

In [78]:
```
test_data = test_data.drop(test_data[test_data['pop']==0].index)
```

In [79]:
```
test_data[test_data['hi_mean'].isna()]
```

Out[79]:

UID	COUNTYID	STATEID	state	state_ab	city	place	type	zip_code	area_code	lat	lng	ALand	AWater
281635	37	47	Tennessee	TN	Nashville	Nashville-davidson Metropolitan Government	City	37209	615	36.179041	-86.883456	14833107	1085155
253631	65	26	Michigan	MI	East Lansing	East Lansing City	CDP	48823	517	42.724476	-84.464366	197880	0
266664	27	36	New York	NY	Poughquag	Hopewell Junction	City	12570	845	41.578838	-73.722796	3206552	62329
270827	119	36	New York	NY	Ossining	Ossining	City	10562	914	41.151312	-73.868465	268098	0
280975	79	45	South Carolina	SC	Columbia	St. Andrews	City	29210	803	34.069566	-81.098423	8774836	666286
224969	29	6	California	CA	Delano	Delano City	City	93215	661	35.783063	-119.312322	2538080	0
282677	157	47	Tennessee	TN	Memphis	Memphis City	City	38107	901	35.053396	-89.971244	19126957	0
226811	37	6	California	CA	Downey	Paramount City	City	90242	562	33.925537	-118.161950	912886	0
259967	51	37	North Carolina	NC	Fort Bragg	Fayetteville City	Village	28307	910	35.124699	-79.015440	5218297	0
222870	13	4	Arizona	AZ	Phoenix	Guadalupe	CDP	85008	602	33.454959	-112.025811	649254	0
285567	201	48	Texas	TX	Houston	West University Place City	Town	77030	713	29.717226	-95.402732	1216880	0
258765	47	28	Mississippi	MS	Gulfport	Gulfport City	CDP	39507	228	30.411360	-89.068583	5014189	6138
264053	25	34	New Jersey	NJ	Tinton Falls	Eatontown	City	7724	732	40.296594	-74.079991	2225754	12029
233123	123	8	Colorado	CO	Greeley	Garden City	City	80631	970	40.402775	-104.701613	627759	0
261031	133	37	North Carolina	NC	Camp Lejuene	Jacksonville City	Village	28547	910	34.664168	-77.342056	19141765	7525847
236210	73	12	Florida	FL	Tallahassee	Tallahassee City	City	32304	850	30.443219	-84.300765	1472129	476

UID	COUNTYID	STATEID	state	state_ab	city	place	type	zip_code	area_code	lat	lng	ALand	AWater
236597	86	12	Florida	FL	Miami	Miami City	City	33132	305	25.778116	-80.193839	297529	0
226187	37	6	California	CA	Diamond Bar	Walnut City	City	91765	909	34.038634	-117.812699	2517842	845
282306	125	47	Tennessee	TN	Woodlawn	Clarksville City	City	37191	931	36.588836	-87.548143	155813940	185782
228308	59	6	California	CA	Anaheim	Garden Grove City	City	92805	714	33.809625	-117.918672	2766486	0
263500	13	34	New Jersey	NJ	Newark	Newark City	City	7105	973	40.696737	-74.156884	16267707	3448139
273184	129	39	Ohio	OH	Grove City	Orient	Village	43123	614	39.794978	-83.148196	5147772	167257
239655	121	13	Georgia	GA	Atlanta	Gresham Park	City	30315	404	33.710292	-84.367788	660642	0
253627	65	26	Michigan	MI	East Lansing	East Lansing City	CDP	48823	517	42.723090	-84.489063	164467	0
287648	453	48	Texas	TX	Austin	Austin City	Town	78703	512	30.305673	-97.760829	503773	0
281150	91	45	South Carolina	SC	Rock Hill	Rock Hill City	City	29730	803	34.939642	-81.030971	416700	0
280990	79	45	South Carolina	SC	Columbia	Dentsville	City	29203	803	34.086856	-80.979093	5462619	165645
249424	71	22	Louisiana	LA	New Orleans	Jefferson	City	70119	504	29.960629	-90.095006	206718	0
238950	53	13	Georgia	GA	Cusseta	Cusseta-chattahoochee	City	31805	706	32.418280	-84.750912	303578182	1283826
251360	3	24	Maryland	MD	Jessup	Jessup	CDP	20794	410	39.135461	-76.777831	2705362	0
286304	245	48	Texas	TX	Port Arthur	Central Gardens	Town	77641	409	29.997881	-94.045169	1592462	25167
257115	163	27	Minnesota	MN	Stillwater	Oak Park Heights City	City	55082	651	45.024958	-92.802094	738408	7791
262260	109	31	Nebraska	NE	Lincoln	Yankee Hill	Village	68512	402	40.770017	-96.703843	194262	0

UID	COUNTYID	STATEID	state	state_ab	city	place	type	zip_code	area_code	lat	lng	ALand	AWater
269337	81	36	New York	NY	Saint Albans	North Valley Stream	City	11412	718	40.688870	-73.770230	524647	0
225038	31	6	California	CA	Corcoran	Corcoran City	City	93212	559	36.057953	-119.554067	5116440	0
259965	51	37	North Carolina	NC	Fort Bragg	Spring Lake	Village	28307	910	35.149896	-79.036151	6302428	54691
232845	59	8	Colorado	CO	Denver	Dakota Ridge	City	80235	303	39.641699	-105.103271	791089	0
260311	77	37	North Carolina	NC	Butner	Butner	Village	27509	919	36.154845	-78.793625	7967277	18933

In [80]:
```
test_data = test_data.drop(test_data[test_data['hi_mean'].isna()].index)
```

In [81]:
```
test_data[rent_na] = test_data.groupby('state')[rent_na].transform(lambda x: x.fillna(x.mean()))
test_data[family_na] = test_data.groupby('state')[family_na].transform(lambda x: x.fillna(x.mean()))
test_data[hc_mortgage_na] = test_data.groupby('state')[hc_mortgage_na].transform(lambda x: x.fillna(x.mean()))
test_data[hc_na] = test_data.groupby('state')[hc_na].transform(lambda x: x.fillna(x.mean()))
```

In [82]:
```
test_data.fillna(test_data.mean(numeric_only = True), inplace = True)
```

In [83]:
```
pd.DataFrame({'Missing Count': test_data.isna().sum(), 'Missing Percent': (test_data.isna().mean().round(4)*100)})
```

Out[83]:

	Missing Count	Missing Percent
COUNTYID	0	0.0
STATEID	0	0.0
state	0	0.0
state_ab	0	0.0
city	0	0.0
place	0	0.0
type	0	0.0
zip_code	0	0.0
area_code	0	0.0
lat	0	0.0
lng	0	0.0
ALand	0	0.0
AWater	0	0.0
pop	0	0.0
male_pop	0	0.0
female_pop	0	0.0
rent_mean	0	0.0
rent_median	0	0.0
rent_stdev	0	0.0
rent_sample_weight	0	0.0
rent_samples	0	0.0
rent_gt_10	0	0.0
rent_gt_15	0	0.0
rent_gt_20	0	0.0

	Missing Count	Missing Percent
rent_gt_25	0	0.0
rent_gt_30	0	0.0
rent_gt_35	0	0.0
rent_gt_40	0	0.0
rent_gt_50	0	0.0
universe_samples	0	0.0
used_samples	0	0.0
hi_mean	0	0.0
hi_median	0	0.0
hi_stdev	0	0.0
hi_sample_weight	0	0.0
hi_samples	0	0.0
family_mean	0	0.0
family_median	0	0.0
family_stdev	0	0.0
family_sample_weight	0	0.0
family_samples	0	0.0
hc_mortgage_mean	0	0.0
hc_mortgage_median	0	0.0
hc_mortgage_stdev	0	0.0
hc_mortgage_sample_weight	0	0.0
hc_mortgage_samples	0	0.0
hc_mean	0	0.0
hc_median	0	0.0

	Missing Count	Missing Percent
hc_stdev	0	0.0
hc_samples	0	0.0
hc_sample_weight	0	0.0
home_equity_second_mortgage	0	0.0
second_mortgage	0	0.0
home_equity	0	0.0
debt	0	0.0
second_mortgage_cdf	0	0.0
home_equity_cdf	0	0.0
debt_cdf	0	0.0
hs_degree	0	0.0
hs_degree_male	0	0.0
hs_degree_female	0	0.0
male_age_mean	0	0.0
male_age_median	0	0.0
male_age_stdev	0	0.0
male_age_sample_weight	0	0.0
male_age_samples	0	0.0
female_age_mean	0	0.0
female_age_median	0	0.0
female_age_stdev	0	0.0
female_age_sample_weight	0	0.0
female_age_samples	0	0.0
pct_own	0	0.0

	Missing Count	Missing Percent
married	0	0.0
married_snp	0	0.0
separated	0	0.0
divorced	0	0.0

In [84]:
```
test_data['Bad_Debt'] = test_data['second_mortgage'] + test_data['home_equity'] - test_data['home_equity_second_mortgage']
```

In [85]:
```
lr_train_data = train_data.drop(category_columns, axis=1)
lr_train_data = lr_train_data.drop(unwanted_columns, axis=1)
```

In [86]:
```
lr_train_data.head()
```

Out[86]:

UID	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_g
267822	5230	2612	2618	769.38638	784.0	232.63967	272.34441	362.0	0.86761	0.79155	0.59155	0.4
246444	2633	1349	1284	804.87924	848.0	253.46747	312.58622	513.0	0.97410	0.93227	0.69920	0.6
245683	6881	3643	3238	742.77365	703.0	323.39011	291.85520	378.0	0.95238	0.88624	0.79630	0.6
279653	2700	1141	1559	803.42018	782.0	297.39258	259.30316	368.0	0.94693	0.87151	0.69832	0.6
247218	5637	2586	3051	938.56493	881.0	392.44096	1005.42886	1704.0	0.99286	0.98247	0.91688	0.8

In [87]:
```
lr_test_data = test_data.drop(category_columns, axis=1)
lr_test_data = lr_test_data.drop(unwanted_columns, axis=1)
```

In [88]:
```
lr_test_data.head()
```

Out[88]:

UID	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_g
255504	3417	1479	1938	858.57169	859.0	232.39082	276.07497	424.0	1.00000	0.95696	0.85316	0.8
252676	3796	1846	1950	832.68625	750.0	267.22342	183.32299	245.0	1.00000	1.00000	0.86611	0.6
276314	3944	2065	1879	816.00639	755.0	416.25699	141.39063	217.0	0.97573	0.93204	0.78641	0.7
248614	2508	1427	1081	418.68937	385.0	156.92024	88.95960	93.0	1.00000	0.93548	0.93548	0.6
286865	6230	3274	2956	1031.63763	997.0	326.76727	277.39844	624.0	0.72276	0.66506	0.53526	0.3

In [89]:
```
x_train = lr_train_data.drop(['hc_mortgage_mean', 'hc_mortgage_median',
       'hc_mortgage_stdev', 'hc_mortgage_sample_weight', 'hc_mortgage_samples'], axis = 1)

y_train = lr_train_data['hc_mortgage_mean']
```

In [90]:
```
x_train.head()
```

Out[90]:

UID	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_g
267822	5230	2612	2618	769.38638	784.0	232.63967	272.34441	362.0	0.86761	0.79155	0.59155	0.4
246444	2633	1349	1284	804.87924	848.0	253.46747	312.58622	513.0	0.97410	0.93227	0.69920	0.6
245683	6881	3643	3238	742.77365	703.0	323.39011	291.85520	378.0	0.95238	0.88624	0.79630	0.6
279653	2700	1141	1559	803.42018	782.0	297.39258	259.30316	368.0	0.94693	0.87151	0.69832	0.6
247218	5637	2586	3051	938.56493	881.0	392.44096	1005.42886	1704.0	0.99286	0.98247	0.91688	0.8

In [91]:
```
y_train.head()
```

Out[91]:
```
UID
267822    1414.80295
246444     864.41390
245683    1506.06758
279653    1175.28642
247218    1192.58759
Name: hc_mortgage_mean, dtype: float64
```

In [92]:
```
x_test = lr_test_data.drop(['hc_mortgage_mean', 'hc_mortgage_median',
       'hc_mortgage_stdev', 'hc_mortgage_sample_weight', 'hc_mortgage_samples'], axis = 1)
y_test = lr_test_data['hc_mortgage_mean']
x_test
```

Out[92]:

UID	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_g
255504	3417	1479	1938	858.57169	859.0	232.39082	276.07497	424.0	1.00000	0.95696	0.85316	0.8
252676	3796	1846	1950	832.68625	750.0	267.22342	183.32299	245.0	1.00000	1.00000	0.86611	0.6
276314	3944	2065	1879	816.00639	755.0	416.25699	141.39063	217.0	0.97573	0.93204	0.78641	0.7
248614	2508	1427	1081	418.68937	385.0	156.92024	88.95960	93.0	1.00000	0.93548	0.93548	0.6
286865	6230	3274	2956	1031.63763	997.0	326.76727	277.39844	624.0	0.72276	0.66506	0.53526	0.3
...	...	...	...	...	...	...	...	...	...	...	...	
238088	5611	2697	2914	1458.82449	1603.0	566.90682	29.43733	99.0	1.00000	1.00000	1.00000	0.6
242811	2695	1504	1191	700.53513	661.0	254.66700	480.86455	592.0	1.00000	0.90034	0.85911	0.6
250127	7392	3669	3723	1069.70567	1138.0	488.13975	207.29615	506.0	0.85375	0.83004	0.77273	0.5
241096	5945	2732	3213	696.93368	576.0	595.16228	503.83775	590.0	0.96886	0.92042	0.83045	0.6
287763	4117	2070	2047	950.09294	864.0	333.82364	417.07457	675.0	1.00000	0.97481	0.86074	0.7

11565 rows × 59 columns

In [93]:
```
print(x_train.shape, x_test.shape)
```

```
(26954, 59) (11565, 59)
```

In [94]:
```
lm.fit(x_train,y_train)
```

Out[94]:
```
LinearRegression()
```

In [95]:
```
lm.coef_
```

Out[95]:
```
array([ 1.93145965e-02,  1.57520885e-01, -1.38206282e-01,  8.35491761e-02,
        7.51755934e-02,  2.52661104e-01, -2.35746122e-01,  2.56615683e-01,
        1.04627026e+02,  3.04194129e+01,  3.79790344e+01,  6.53512696e+01,
        6.31203875e+01,  2.96127401e+00, -3.70453298e+01, -4.31675913e+00,
        4.58190445e-01, -3.80185357e-01,  1.82731346e-03,  1.67166036e-03,
        7.41970460e-03,  7.72565206e-02, -3.51564249e-01,  1.34569556e-03,
        7.10154204e-04,  1.32554826e-03, -1.95520413e-01,  2.99689810e-01,
        6.72412182e-01,  1.81019757e-01,  2.92452146e-01, -1.39275809e-01,
        3.89538206e-01, -5.77605384e+02,  6.06164648e+02, -2.93445203e+02,
       -2.62280845e+02,  1.66529106e+01, -7.47913372e+01, -2.53567046e+02,
        3.01008436e+02, -2.83502542e+02, -5.56396536e+02,  3.32638478e+00,
        4.33336715e+00,  3.93130633e+00, -5.69546830e-02, -1.45108728e-01,
       -5.72902420e-01,  2.94008122e+00, -2.32344154e+00,  1.12055532e-01,
        1.13444076e-01, -4.25626493e+02, -1.84377696e+02,  3.46921181e+02,
       -4.59152252e+02, -5.80339355e+02,  8.90324829e+02])
```

In [96]:
```
coeff_df = pd.DataFrame(lm.coef_ , index = x_train.columns , columns = ['Coefficient'])
coeff_df.round(4)
```

Out[96]:

	Coefficient
pop	0.0193
male_pop	0.1575
female_pop	-0.1382
rent_mean	0.0835
rent_median	0.0752
rent_stdev	0.2527
rent_sample_weight	-0.2357
rent_samples	0.2566
rent_gt_10	104.6270
rent_gt_15	30.4194
rent_gt_20	37.9790
rent_gt_25	65.3513
rent_gt_30	63.1204
rent_gt_35	2.9613
rent_gt_40	-37.0453
rent_gt_50	-4.3168
universe_samples	0.4582
used_samples	-0.3802
hi_mean	0.0018
hi_median	0.0017
hi_stdev	0.0074
hi_sample_weight	0.0773
hi_samples	-0.3516
family_mean	0.0013

	Coefficient
family_median	0.0007
family_stdev	0.0013
family_sample_weight	-0.1955
family_samples	0.2997
hc_mean	0.6724
hc_median	0.1810
hc_stdev	0.2925
hc_samples	-0.1393
hc_sample_weight	0.3895
home_equity_second_mortgage	-577.6054
second_mortgage	606.1646
home_equity	-293.4452
debt	-262.2808
second_mortgage_cdf	16.6529
home_equity_cdf	-74.7913
debt_cdf	-253.5670
hs_degree	301.0084
hs_degree_male	-283.5025
hs_degree_female	-556.3965
male_age_mean	3.3264
male_age_median	4.3334
male_age_stdev	3.9313
male_age_sample_weight	-0.0570
male_age_samples	-0.1451

	Coefficient
female_age_mean	-0.5729
female_age_median	2.9401
female_age_stdev	-2.3234
female_age_sample_weight	0.1121
female_age_samples	0.1134
pct_own	-425.6265
married	-184.3777
married_snp	346.9212
separated	-459.1523
divorced	-580.3394
Bad_Debt	890.3248

In [97]:
```
lm.intercept_
```

Out[97]:
```
505.61964283750603
```

In [98]:
```
predictions = lm.predict(x_test)
```

In [99]:
```
predictions
```

Out[99]:
```
array([1174.68133217, 1500.60054286, 1209.86188323, ..., 1793.37061446,
       1172.35887029, 1479.35390991])
```

In [100…
```
pred_data = pd.DataFrame({'Original' : y_test , 'Prediction by Model' : predictions})
pred_data.round(2)
```

Out[100]:

UID	Original	Prediction by Model
255504	1139.25	1174.68
252676	1533.26	1500.60
276314	1254.54	1209.86
248614	862.66	798.96
286865	1996.41	2166.90
...	...	...
238088	1269.83	1443.96
242811	1406.83	1336.48
250127	1791.64	1793.37
241096	1182.30	1172.36
287763	1364.17	1479.35

11565 rows × 2 columns

In [101…
```
from sklearn.metrics import mean_absolute_error, mean_squared_error , mean_absolute_percentage_error , r2_score
```

In [102…
```
mae = mean_absolute_error(y_test, predictions).round(4)
mse = mean_squared_error(y_test, predictions).round(4)
rmse = np.sqrt(mean_squared_error(y_test, predictions)).round(4)
mape = mean_absolute_percentage_error(y_test, predictions).round(4)
r2 = r2_score(y_test, predictions).round(4)
```

In [103…
```
print('Mean Absolute Error (MAE):', mae)
print('Mean Squared Error (MSE):', mse)
print('Root Mean Squared Error (RMSE):', rmse)
print('Mean Absolute Percentage Error (MAPE):', mape)
print('R2:', r2)
```

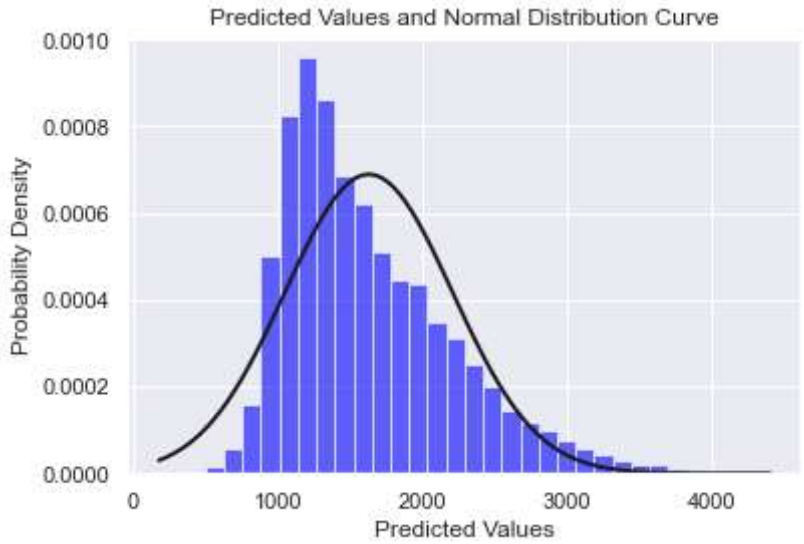
```
Mean Absolute Error (MAE): 174.9382
Mean Squared Error (MSE): 61938.9373
Root Mean Squared Error (RMSE): 248.8753
Mean Absolute Percentage Error (MAPE): 0.1117
R2: 0.8452
```

In [104…
```
n = x_train.shape[0]
p = x_train.shape[1]
```

In [105…
```
adj_r2 = 1 - (1 - r2_score(y_test, predictions)) * (n - 1) / (n - p - 1)
adj_r2.round(4)
```

Out[105]:
```
0.8449
```

In [106…
```
plt.hist(predictions, bins=30, density=True, alpha=0.6, color='blue')
x_min, x_max = plt.xlim()
x = np.linspace(x_min, x_max, 100)
p = norm.pdf(x, np.mean(predictions), np.std(predictions))
plt.plot(x, p, 'k', linewidth=2)
plt.xlabel('Predicted Values')
plt.ylabel('Probability Density')
plt.title('Predicted Values and Normal Distribution Curve')
plt.show()
```



The predicted values are positively skewed. They do not appear to be normally distributed. However, to check whether they are normally distributed using a statistical measure, Shapiro-Wilk test and Anderson-Darling tests are performed.

Shapiro-Wilk test

In [107…
```
from scipy.stats import shapiro

# perform the Shapiro-Wilk test
stat, pvalue = shapiro(predictions)
print("Statistic:", stat, "Pvalue:", pvalue)
# interpret the test results
alpha = 0.05

if pvalue > alpha:
    print('The data is normally distributed')
else:
    print('The data is not normally distributed')
```

```
Statistic: 0.9321021437644958 Pvalue: 0.0
The data is not normally distributed
C:\Users\14sac\anaconda3\lib\site-packages\scipy\stats\morestats.py:1760: UserWarning: p-value may not be accurate for N > 5000.
  warnings.warn("p-value may not be accurate for N > 5000.")
```

Anderson-Darling test

In [108…
```
from scipy.stats import anderson
```

In [109…
```
anderson(predictions)
```

Out[109]:
```
AndersonResult(statistic=226.12719720747555, critical_values=array([0.576, 0.656, 0.787, 0.918, 1.092]), significance_level=arra
y([15. , 10. ,  5. ,  2.5,  1. ]))
```

The critical value for α = 0.025 is 0.858. Because the test statistic i.e. 226.127 is greater than all the critical values (for different levels of significance), the results are significant at all significance levels. Hence, the null hypothesis can be rejected. i.e. there is sufficient evidence to say that the data is not normally distributed.

In [ ]: