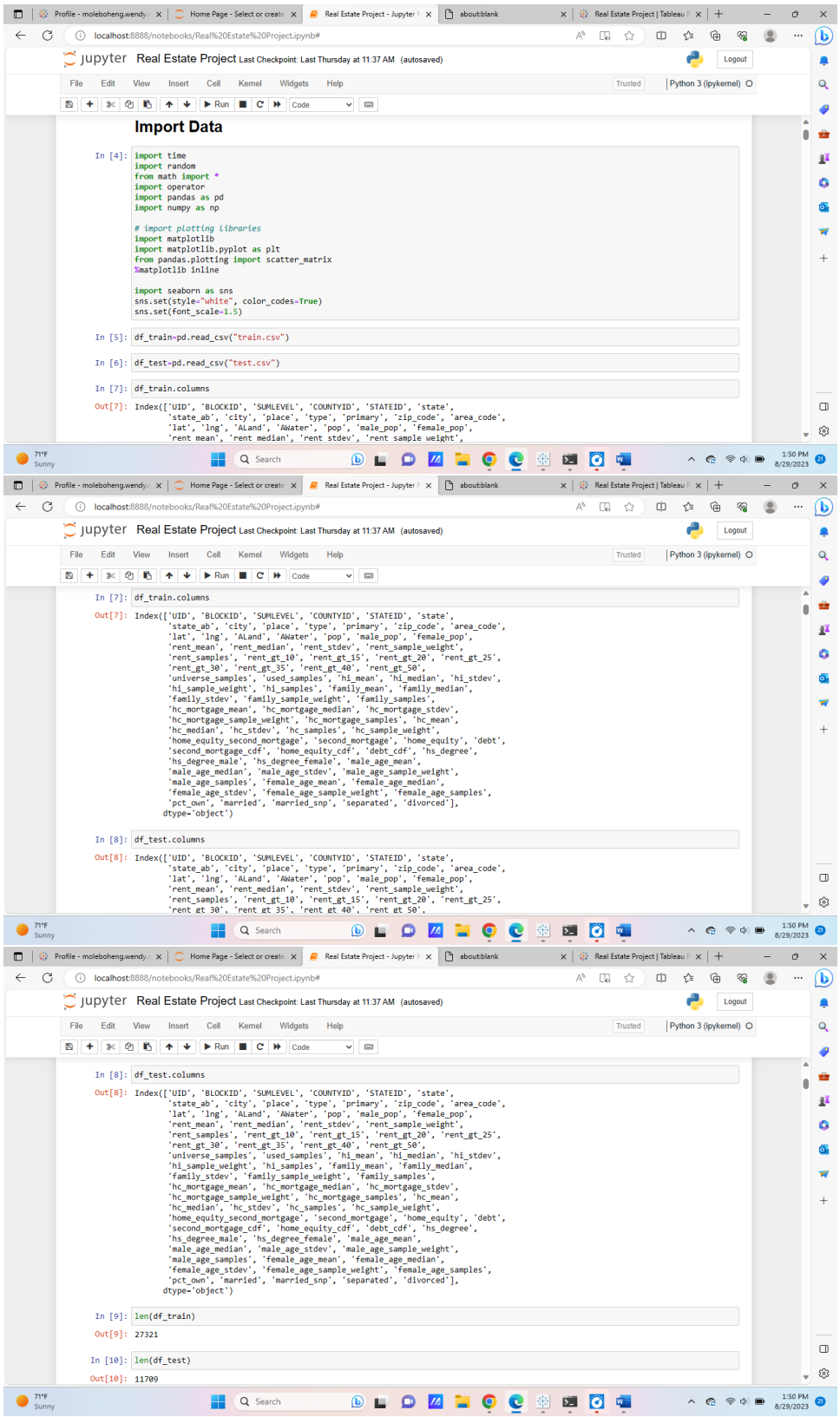
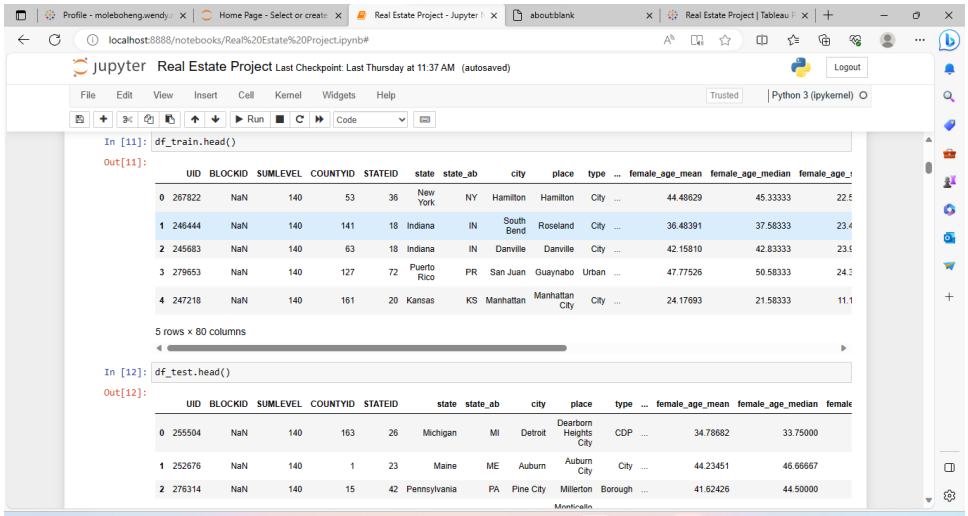
Real Estate Capstone

Course-end Project 1





jupyter Real Estate Project Last Checkpoint: Last Thursday at 11:37 AM (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

In [11]:
```
df_train.head()
```

Out[11]:

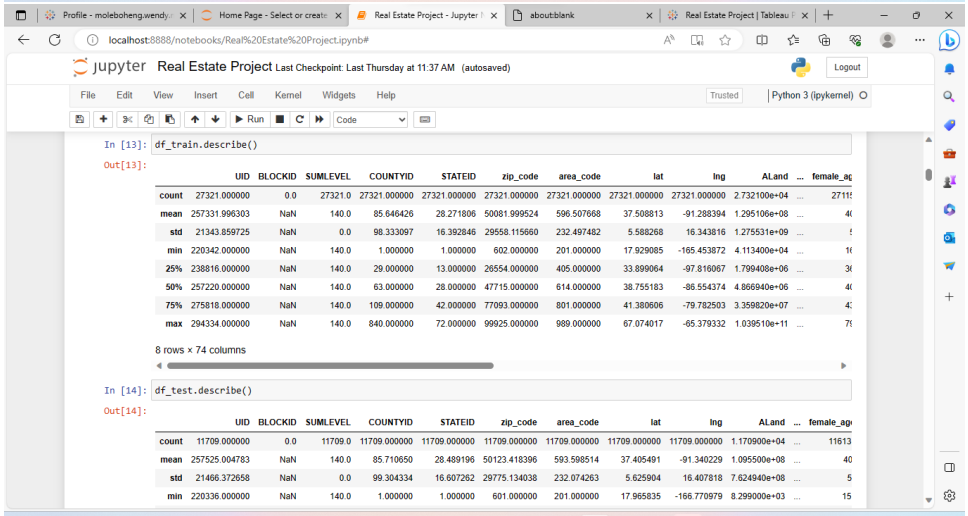
	UID	BLOCKID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	type	...	female_age_mean	female_age_median	female_age_s
0	267822	NaN	140	53	36	New York	NY	Hamilton	Hamilton	City	...	44.48629	45.33333	22.5
1	246444	NaN	140	141	18	Indiana	IN	South Bend	Roseland	City	...	36.48391	37.58333	23.4
2	245683	NaN	140	63	18	Indiana	IN	Danville	Danville	City	...	42.15810	42.83333	23.5
3	279653	NaN	140	127	72	Puerto Rico	PR	San Juan	Guaynabo	Urban	...	47.77526	50.58333	24.3
4	247218	NaN	140	161	20	Kansas	KS	Manhattan	Manhattan City	City	...	24.17693	21.58333	11.1

5 rows × 80 columns

In [12]:
```
df_test.head()
```

Out[12]:

	UID	BLOCKID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	type	...	female_age_mean	female_age_median	female
0	255504	NaN	140	163	26	Michigan	MI	Detroit	Dearborn Heights City	CDP	...	34.78682	33.75000	
1	252676	NaN	140	1	23	Maine	ME	Auburn	Auburn City	City	...	44.23451	46.66667	
2	276314	NaN	140	15	42	Pennsylvania	PA	Pine City	Millerton	Borough	...	41.62426	44.50000	



jupyter Real Estate Project Last Checkpoint: Last Thursday at 11:37 AM (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

In [13]:
```
df_train.describe()
```

Out[13]:

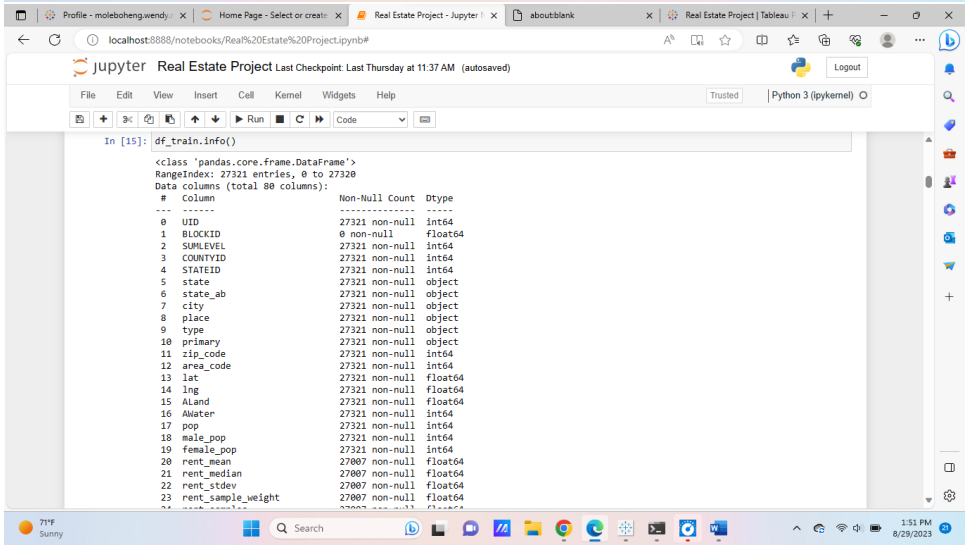
	UID	BLOCKID	SUMLEVEL	COUNTYID	STATEID	zip_code	area_code	lat	lng	ALand	...	female_ag
count	27321.000000	0.0	27321.0	27321.000000	27321.000000	27321.000000	27321.000000	27321.000000	27321.000000	2.732100e+04	...	27115
mean	257331.996303	NaN	140.0	85.646426	28.271806	50081.999524	596.507668	37.508813	-91.288394	1.295106e+08	...	40
std	21343.859725	NaN	0.0	98.333097	16.392846	29558.115660	232.497482	5.588268	16.343816	1.275531e+09	...	5
min	220342.000000	NaN	140.0	1.000000	1.000000	602.000000	201.000000	17.929085	-165.453872	4.113400e+04	...	16
25%	238816.000000	NaN	140.0	29.000000	13.000000	26554.000000	405.000000	33.899064	-97.816067	1.799408e+06	...	36
50%	257220.000000	NaN	140.0	63.000000	28.000000	47715.000000	614.000000	38.755183	-86.554374	4.866940e+06	...	40
75%	275818.000000	NaN	140.0	109.000000	42.000000	77093.000000	801.000000	41.380606	-79.782503	3.359820e+07	...	43
max	294334.000000	NaN	140.0	840.000000	72.000000	99925.000000	989.000000	67.074017	-65.379332	1.039510e+11	...	75

8 rows × 74 columns

In [14]:
```
df_test.describe()
```

Out[14]:

	UID	BLOCKID	SUMLEVEL	COUNTYID	STATEID	zip_code	area_code	lat	lng	ALand	...	female_ag
count	11709.000000	0.0	11709.0	11709.000000	11709.000000	11709.000000	11709.000000	11709.000000	11709.000000	1.170900e+04	...	11613
mean	257525.004783	NaN	140.0	85.710650	28.489196	50123.418396	593.598514	37.405491	-91.340229	1.095500e+08	...	40
std	21466.372658	NaN	0.0	99.304334	16.607262	29775.134038	232.074263	5.625904	16.407818	7.624940e+08	...	5
min	220336.000000	NaN	140.0	1.000000	1.000000	601.000000	201.000000	17.965835	-166.770979	8.299000e+03	...	15



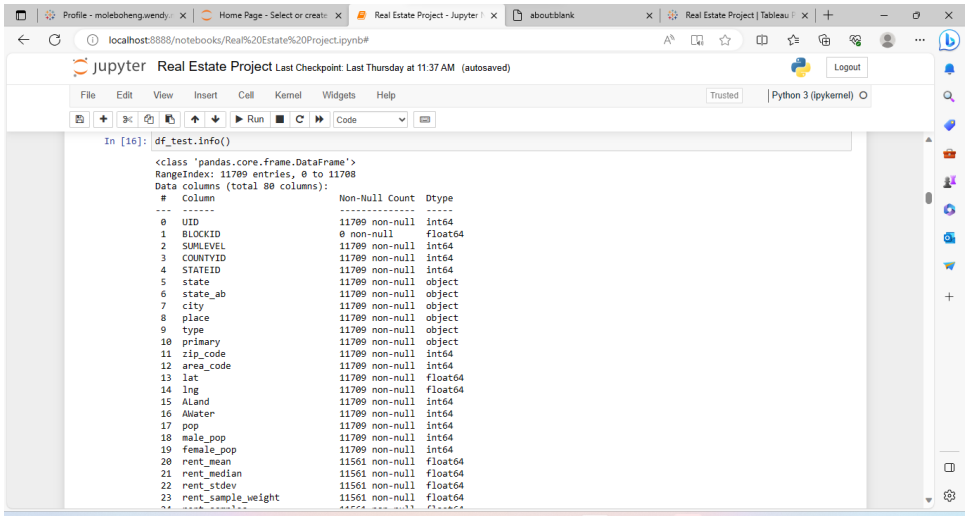
jupyter Real Estate Project Last Checkpoint: Last Thursday at 11:37 AM (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

In [15]:
```
df_train.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
```



```
In [16]: 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
```

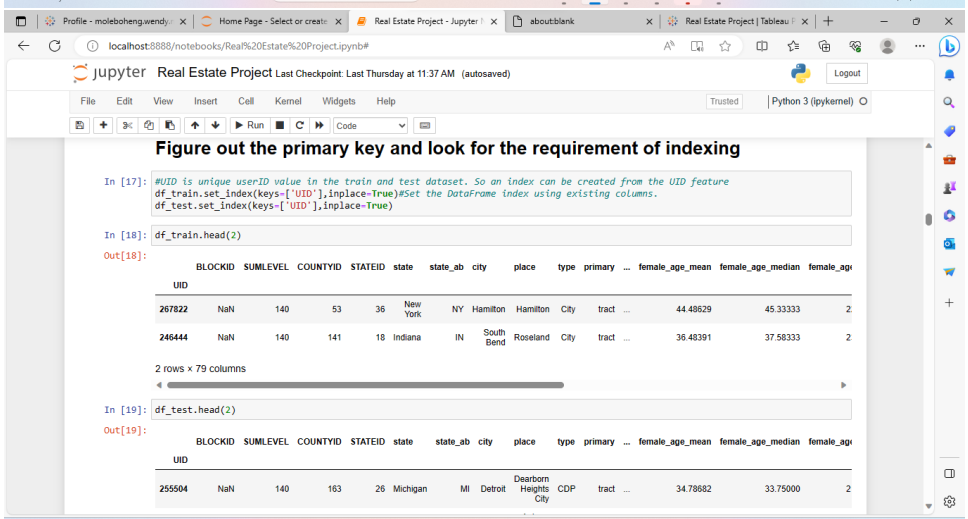


Figure out the primary key and look for the requirement of indexing

```
In [17]: #UID is unique userID value in the train and test dataset. So an index can be created from the UID feature
df_train.set_index(keys=['UID'],inplace=True)#Set the DataFrame index using existing columns.
df_test.set_index(keys=['UID'],inplace=True)

In [18]: df_train.head(2)
```

Out[18]:

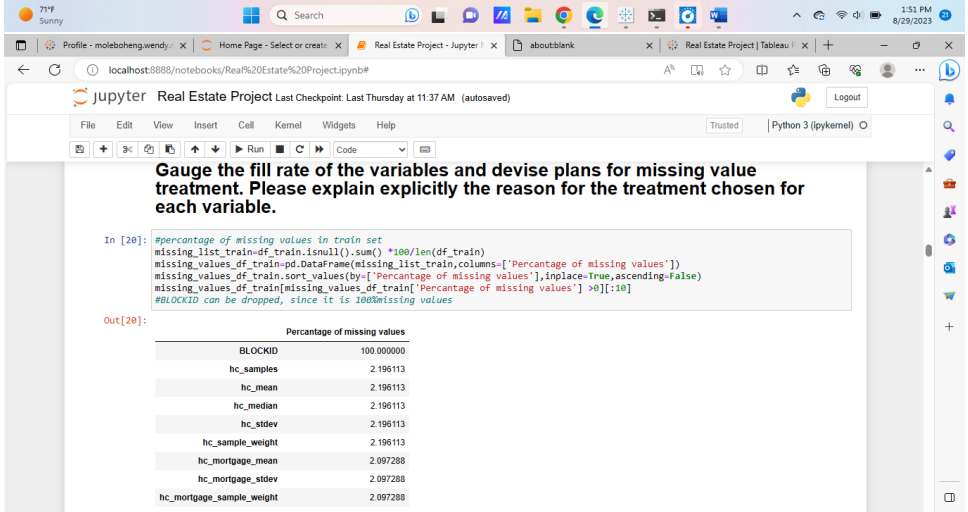
UID	BLOCKID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	type	primary	...	female_age_mean	female_age_median	female_age
267822	NaN	140	53	36	New York	NY	Hamilton	Hamilton	City	tract	...	44.48629	45.33333	2
246444	NaN	140	141	18	Indiana	IN	South Bend	Roseland	City	tract	...	36.48391	37.58333	2

2 rows × 79 columns

```
In [19]: df_test.head(2)
```

Out[19]:

UID	BLOCKID	SUMLEVEL	COUNTYID	STATEID	state	state_ab	city	place	type	primary	...	female_age_mean	female_age_median	female_age
255504	NaN	140	163	26	Michigan	MI	Detroit	Dearborn Heights City	CDP	tract	...	34.78682	33.75000	2

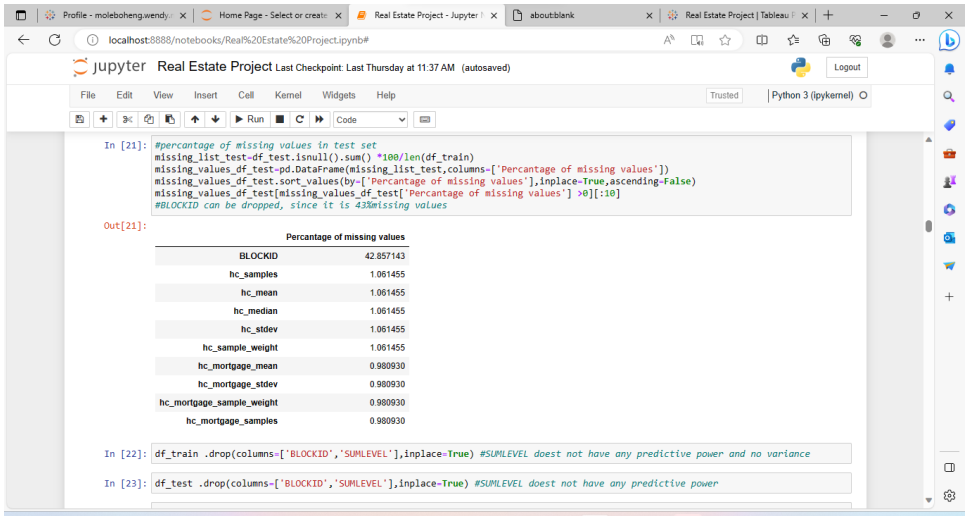


Gauge the fill rate of the variables and devise plans for missing value treatment. Please explain explicitly the reason for the treatment chosen for each variable.

```
In [20]: #percantage of missing values in train set
missing_list_train=df_train.isnull().sum() *100/len(df_train)
missing_values_df_train=pd.DataFrame(missing_list_train,columns=['Percantage of missing values'])
missing_values_df_train.sort_values(by=['Percantage of missing values'],inplace=True,ascending=False)
missing_values_df_train[missing_values_df_train['Percantage of missing values'] >0][:10]
#BLOCKID can be dropped, since it is 100%missing values
```

Out[20]:

	Percantage of missing values
BLOCKID	100.000000
hc_samples	2.196113
hc_mean	2.196113
hc_median	2.196113
hc_stdev	2.196113
hc_sample_weight	2.196113
hc_mortgage_mean	2.097288
hc_mortgage_stdev	2.097288
hc_mortgage_sample_weight	2.097288
hc_mortgage_samples	2.097288



jupyter Real Estate Project Last Checkpoint: Last Thursday at 11:37 AM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

In [21]:
```
#percentage of missing values in test set
missing_list_test=df_test.isnull().sum() *100/len(df_train)
missing_values_df_test=pd.DataFrame(missing_list_test,columns=['Percantage of missing values'])
missing_values_df_test.sort_values(by=['Percantage of missing values'],inplace=True,ascending=False)
missing_values_df_test[missing_values_df_test['Percantage of missing values'] >0][:10]
#BLOCKID can be dropped, since it is 43%missing values
```

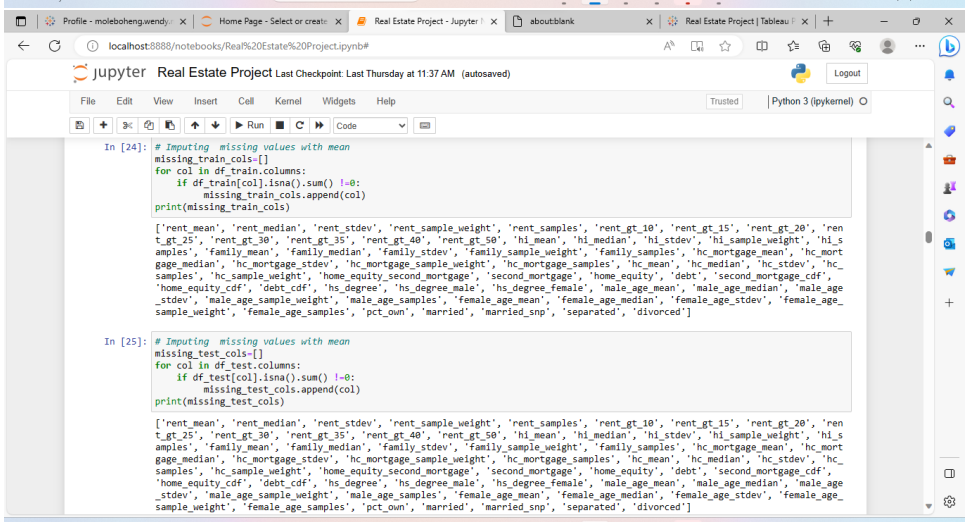
Out[21]:

	Percantage of missing values
BLOCKID	42.857143
hc_samples	1.061455
hc_mean	1.061455
hc_median	1.061455
hc_stdev	1.061455
hc_sample_weight	1.061455
hc_mortgage_mean	0.980930
hc_mortgage_stdev	0.980930
hc_mortgage_sample_weight	0.980930
hc_mortgage_samples	0.980930

In [22]:
```
df_train .drop(columns=['BLOCKID','SUMLEVEL'],inplace=True) #SUMLEVEL doest not have any predictive power and no variance
```

In [23]:
```
df_test .drop(columns=['BLOCKID','SUMLEVEL'],inplace=True) #SUMLEVEL doest not have any predictive power
```

71°F Sunny | Search | 1:51 PM 8/29/2023



jupyter Real Estate Project Last Checkpoint: Last Thursday at 11:37 AM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

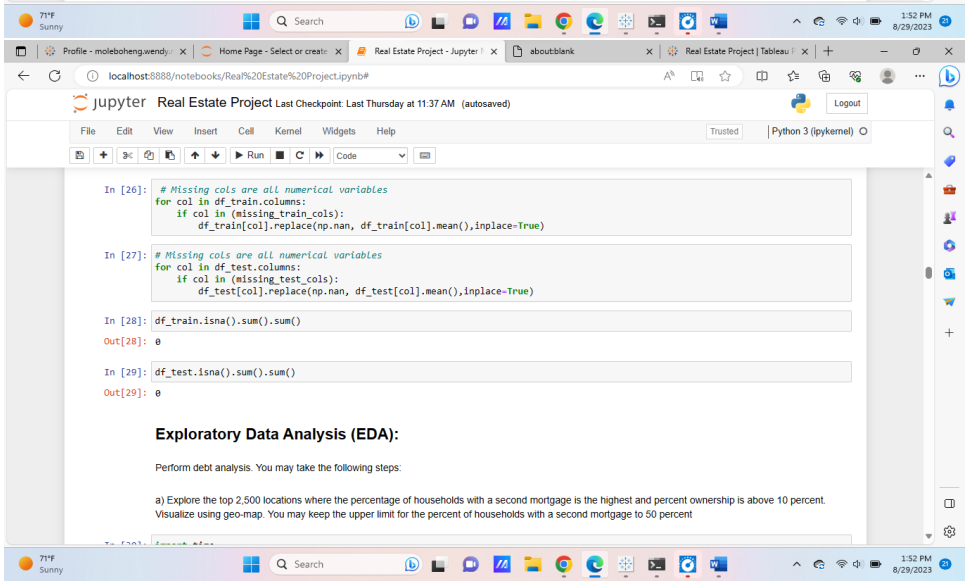
In [24]:
```
# Imputing  missing values with mean
missing_train_cols=[]
for col in df_train.columns:
    if df_train[col].isna().sum() !=0:
        missing_train_cols.append(col)
print(missing_train_cols)
```

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

In [25]:
```
# Imputing  missing values with mean
missing_test_cols=[]
for col in df_test.columns:
    if df_test[col].isna().sum() !=0:
        missing_test_cols.append(col)
print(missing_test_cols)
```

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

71°F Sunny | Search | 1:52 PM 8/29/2023



jupyter Real Estate Project Last Checkpoint: Last Thursday at 11:37 AM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

In [26]:
```
# Missing cols are all numerical variables
for col in df_train.columns:
    if col in (missing_train_cols):
        df_train[col].replace(np.nan, df_train[col].mean(),inplace=True)
```

In [27]:
```
# Missing cols are all numerical variables
for col in df_test.columns:
    if col in (missing_test_cols):
        df_test[col].replace(np.nan, df_test[col].mean(),inplace=True)
```

In [28]:
```
df_train.isna().sum().sum()
```

Out[28]: 0

In [29]:
```
df_test.isna().sum().sum()
```

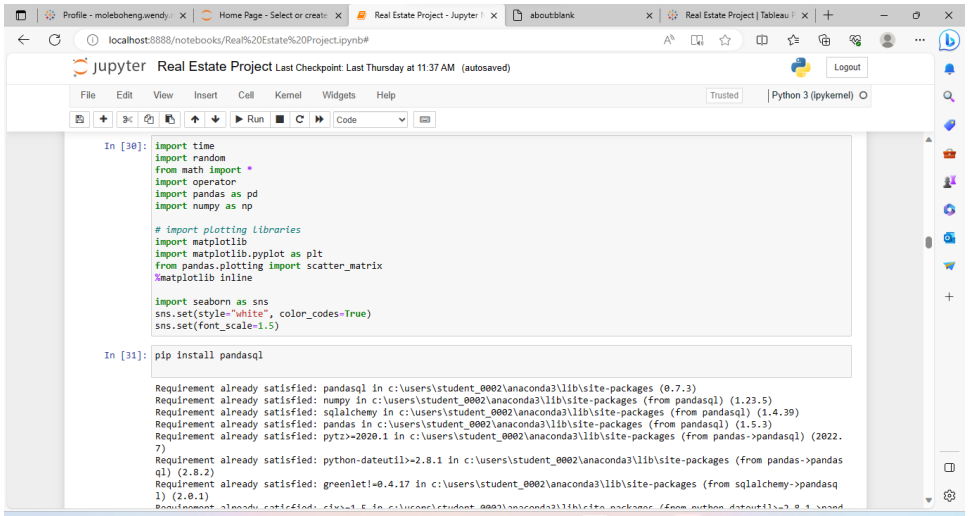
Out[29]: 0

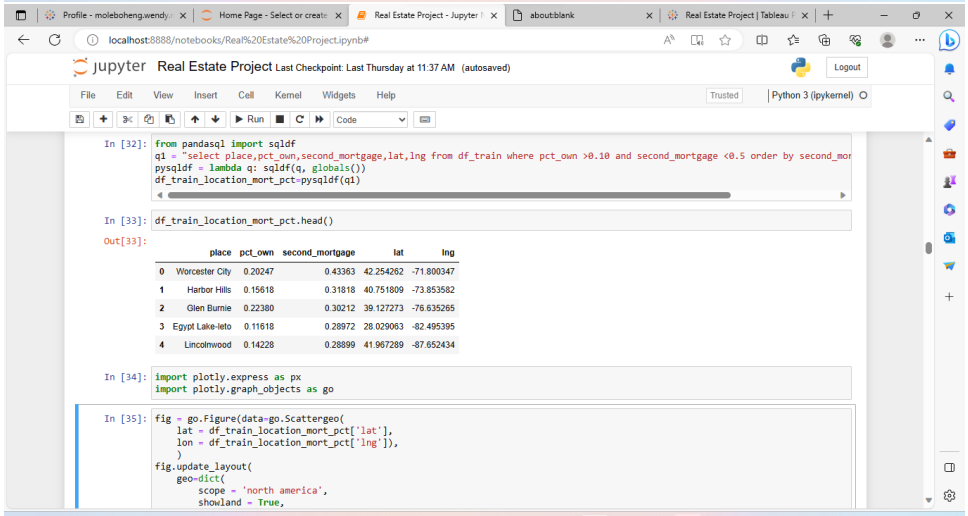
Exploratory Data Analysis (EDA):

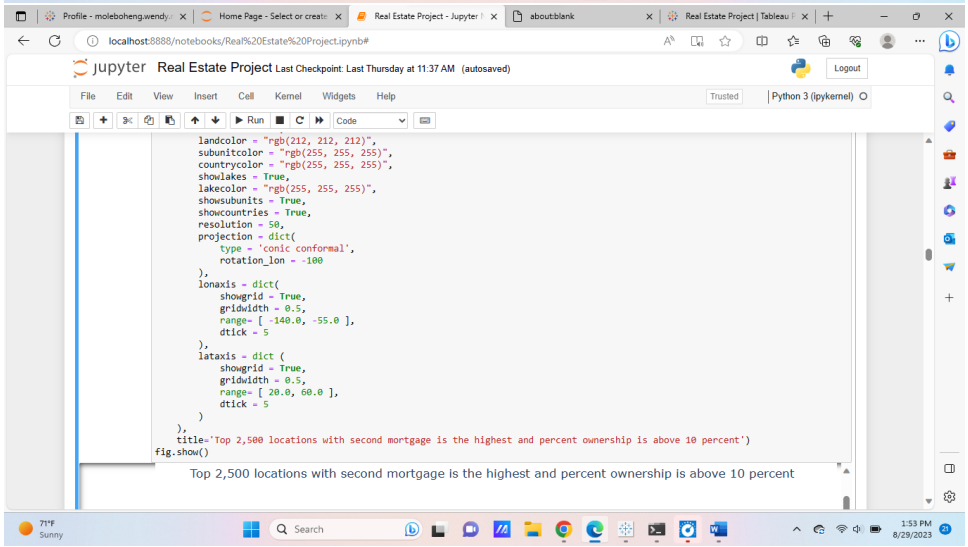
Perform debt analysis. You may take the following steps:

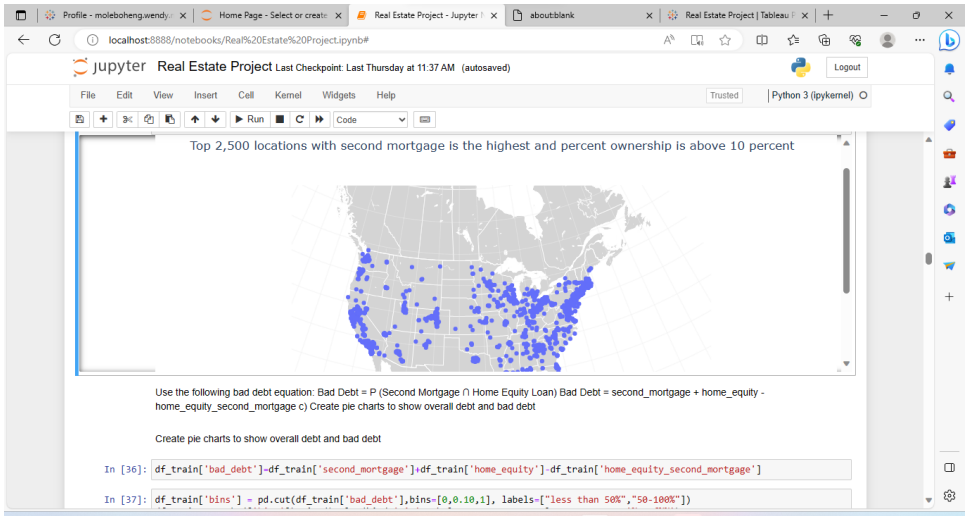
a) Explore the top 2,500 locations where the percentage of households with a second mortgage is the highest and percent ownership is above 10 percent. Visualize using geo-map. You may keep the upper limit for the percent of households with a second mortgage to 50 percent

71°F Sunny | Search | 1:52 PM 8/29/2023


jupyter Real Estate Project Last Checkpoint: Last Thursday at 11:37 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [30]:
import time
import random
from math import *
import operator
import pandas as pd
import numpy as np
# import plotting libraries
import matplotlib
import matplotlib.pyplot as plt
from pandas.plotting import scatter_matrix
%matplotlib inline
import seaborn as sns
sns.set(style="white", color_codes=True)
sns.set(font_scale=1.5)
In [31]:
pip install pandasql
Requirement already satisfied: pandasql in c:\users\student_0002\anaconda3\lib\site-packages (0.7.3)
Requirement already satisfied: numpy in c:\users\student_0002\anaconda3\lib\site-packages (from pandasql) (1.23.5)
Requirement already satisfied: sqlalchemy in c:\users\student_0002\anaconda3\lib\site-packages (from pandasql) (1.4.39)
Requirement already satisfied: pandas in c:\users\student_0002\anaconda3\lib\site-packages (from pandasql) (1.5.3)
Requirement already satisfied: pytz>=2020.1 in c:\users\student_0002\anaconda3\lib\site-packages (from pandas->pandasql) (2022.7)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\student_0002\anaconda3\lib\site-packages (from pandas->pandasql) (2.8.2)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\student_0002\anaconda3\lib\site-packages (from sqlalchemy->pandasql) (2.0.1)


jupyter Real Estate Project Last Checkpoint: Last Thursday at 11:37 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [32]:
from pandasql import sqldf
q1 = "select place,pct_own,second_mortgage,lat,lng from df_train where pct_own >0.10 and second_mortgage <0.5 order by second_mor
pysqldf = lambda q: sqldf(q, globals())
df_train_location_mort_pct=pysqldf(q1)
In [33]:
df_train_location_mort_pct.head()
Out[33]:
place pct_own second_mortgage lat lng
0 Worcester City 0.20247 0.43363 42.254262 -71.800347
1 Harbor Hills 0.15618 0.31818 40.751809 -73.853582
2 Glen Burnie 0.22380 0.30212 39.127273 -76.635265
3 Egypt Lake-leto 0.11618 0.28972 28.029063 -82.495395
4 Lincolnwood 0.14228 0.28899 41.967289 -87.652434
In [34]:
import plotly.express as px
import plotly.graph_objects as go
In [35]:
fig = go.Figure(data=go.Scattergeo(
lat = df_train_location_mort_pct['lat'],
lon = df_train_location_mort_pct['lng']),
)
fig.update_layout(
geo=dict(
scope = 'north america',
showland = True,


jupyter Real Estate Project Last Checkpoint: Last Thursday at 11:37 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
landcolor = "rgb(212, 212, 212)",
subunitcolor = "rgb(255, 255, 255)",
countrycolor = "rgb(255, 255, 255)",
showlakes = True,
lakecolor = "rgb(255, 255, 255)",
showsubunits = True,
showcountries = True,
resolution = 50,
projection = dict(
type = 'conic conformal',
rotation_lon = -100
),
lonaxis = dict(
showgrid = True,
gridwidth = 0.5,
range= [-140.0, -55.0],
dtick = 5
),
lataxis = dict (
showgrid = True,
gridwidth = 0.5,
range= [20.0, 60.0],
dtick = 5
)
),
title='Top 2,500 locations with second mortgage is the highest and percent ownership is above 10 percent')
fig.show()
Top 2,500 locations with second mortgage is the highest and percent ownership is above 10 percent

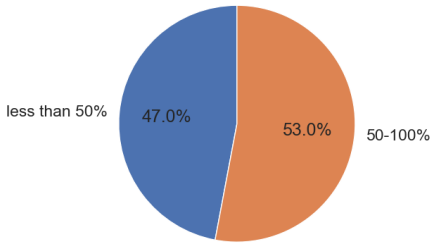


Use the following bad debt equation: Bad Debt = P (Second Mortgage ∩ Home Equity Loan) Bad Debt = second_mortgage + home_equity - home_equity_second_mortgage c) Create pie charts to show overall debt and bad debt

Create pie charts to show overall debt and bad debt

```
In [36]: df_train['bad_debt']=df_train['second_mortgage']+df_train['home_equity']-df_train['home_equity_second_mortgage']

In [37]: df_train['bins'] = pd.cut(df_train['bad_debt'],bins=[0,0.10,1], labels=["less than 50%","50-100%"])
```

```
In [37]: df_train['bins'] = pd.cut(df_train['bad_debt'],bins=[0,0.10,1], labels=["less than 50%","50-100%"])
         df_train.groupby(['bins']).size().plot(kind='pie',subplots=True,startangle=90, autopct='%1.1f%%')
         plt.axis('equal')

         plt.show()
         #df.plot.pie(subplots=True,figsize=(8, 3))
```



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

```
In [38]: cols=[]
         df_train.columns

Out[38]: Index(['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', 'bins'],
               dtype='object')

In [39]: #Taking Hamilton and Manhattan cities data
         cols=['second_mortgage','home_equity','debt','bad_debt']
         df_box_hamilton=df_train.loc[df_train['city'] == 'Hamilton']
         df_box_manhattan=df_train.loc[df_train['city'] == 'Manhattan']
```

jupyter Real Estate Project Last Checkpoint: Last Thursday at 11:37 AM (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

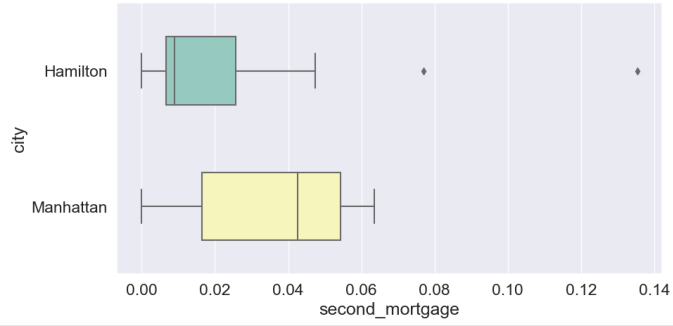
In [39]:
```
#Taking Hamilton and Manhattan cities data
cols=['second_mortgage','home_equity','debt','bad_debt']
df_box_hamilton=df_train.loc[df_train['city'] == 'Hamilton']
df_box_manhattan=df_train.loc[df_train['city'] == 'Manhattan']
df_box_city=pd.concat([df_box_hamilton,df_box_manhattan])
df_box_city.head(4)
```

Out[39]:

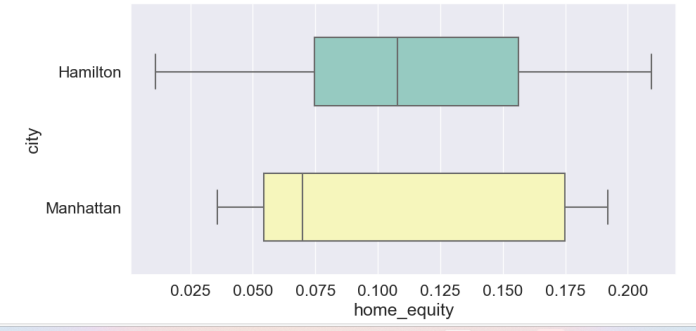
UID	COUNTYID	STATEID	state	state_ab	city	place	type	primary	zip_code	area_code	...	female_age_stdev	female_age_sample_weight
267822	53	36	New York	NY	Hamilton	Hamilton	City	tract	13346	315	...	22.51276	685.33845
263797	21	34	New Jersey	NJ	Hamilton	Yardville	City	tract	8610	609	...	24.05831	732.58443
270979	17	39	Ohio	OH	Hamilton	Hamilton City	Village	tract	45015	513	...	22.66500	565.32725
259028	95	28	Mississippi	MS	Hamilton	Hamilton	CDP	tract	39746	662	...	22.79602	483.01311

4 rows × 79 columns

In [40]:
```
plt.figure(figsize=(10,5))
sns.boxplot(data=df_box_city,x='second_mortgage', y='city',width=0.5,palette="Set3")
plt.show()
```



In [41]:
```
plt.figure(figsize=(10,5))
sns.boxplot(data=df_box_city,x='home_equity', y='city',width=0.5,palette="Set3")
plt.show()
```



jupyter Real Estate Project Last Checkpoint: Last Thursday at 11:37 AM (autosaved)

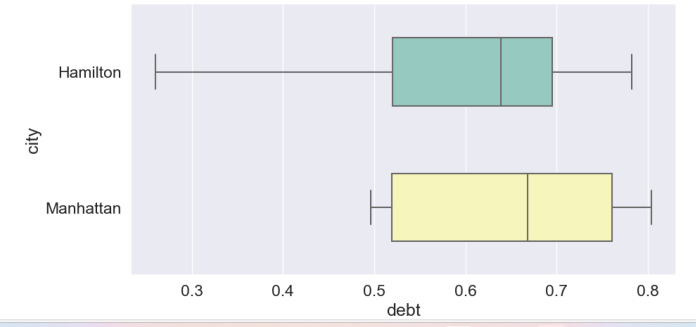
Logout

File Edit View Insert Cell Kernel Widgets Help

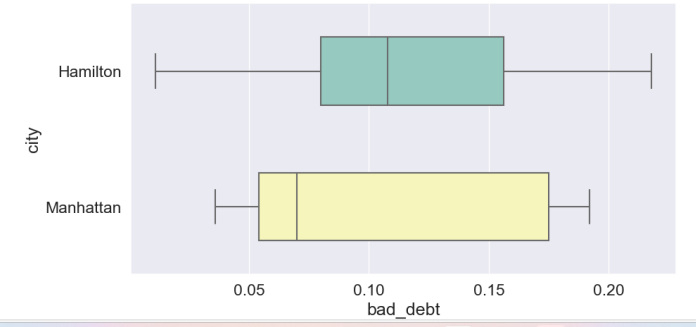
Trusted | Python 3 (ipykernel)

Run Code

In [42]:
```
plt.figure(figsize=(10,5))
sns.boxplot(data=df_box_city,x='debt', y='city',width=0.5,palette="Set3")
plt.show()
```



In [43]:
```
plt.figure(figsize=(10,5))
sns.boxplot(data=df_box_city,x='bad_debt', y='city',width=0.5,palette="Set3")
plt.show()
```

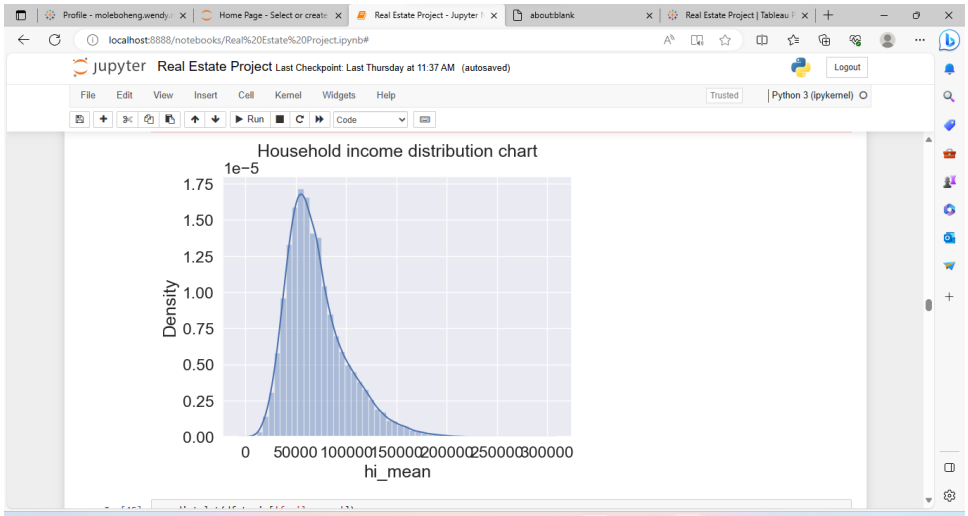


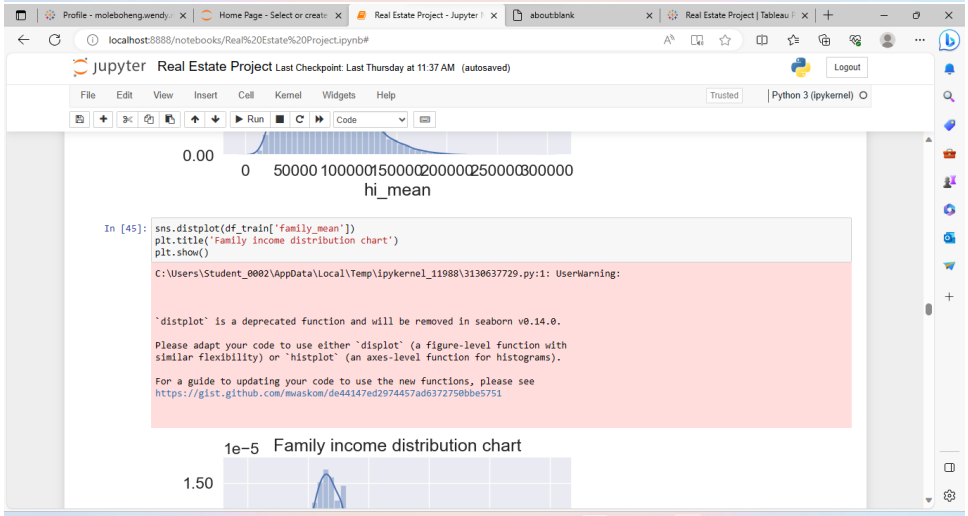
Manhattan has higher metrics compared to Hamilton

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

In [44]:
```
sns.distplot(df_train['hi_mean'])
plt.title('Household income distribution chart')
plt.show()
```

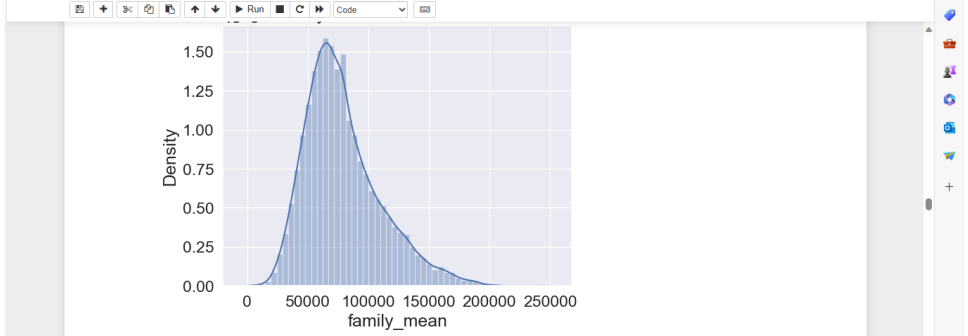
```
C:\Users\Student_0002\AppData\Local\Temp\ipykernel_11988\2321983864.py:1: UserWarning:


`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
```



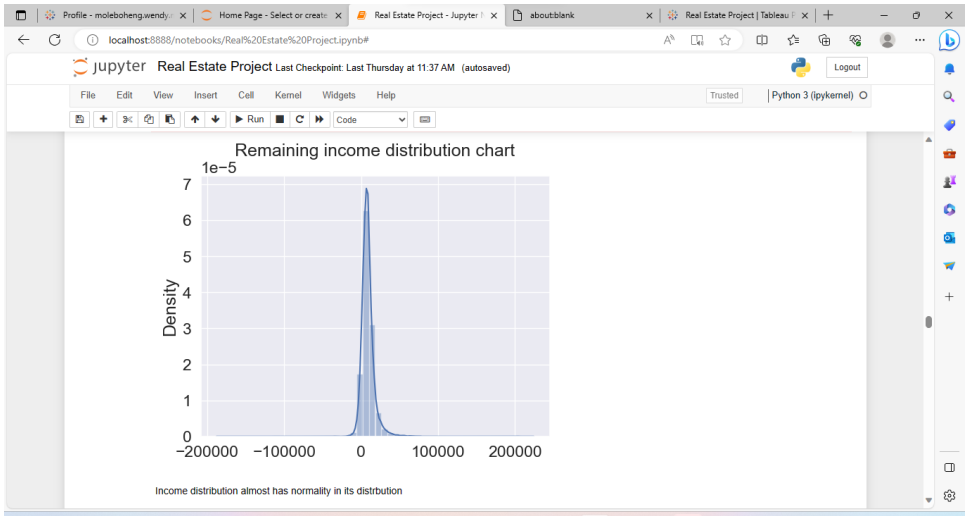


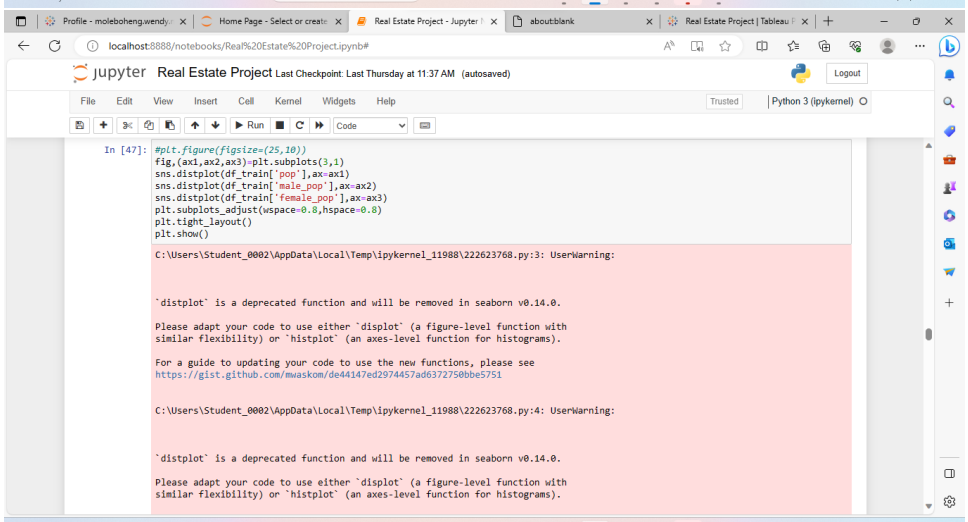

```
In [45]: sns.distplot(df_train['family_mean'])
         plt.title('Family income distribution chart')
         plt.show()
```

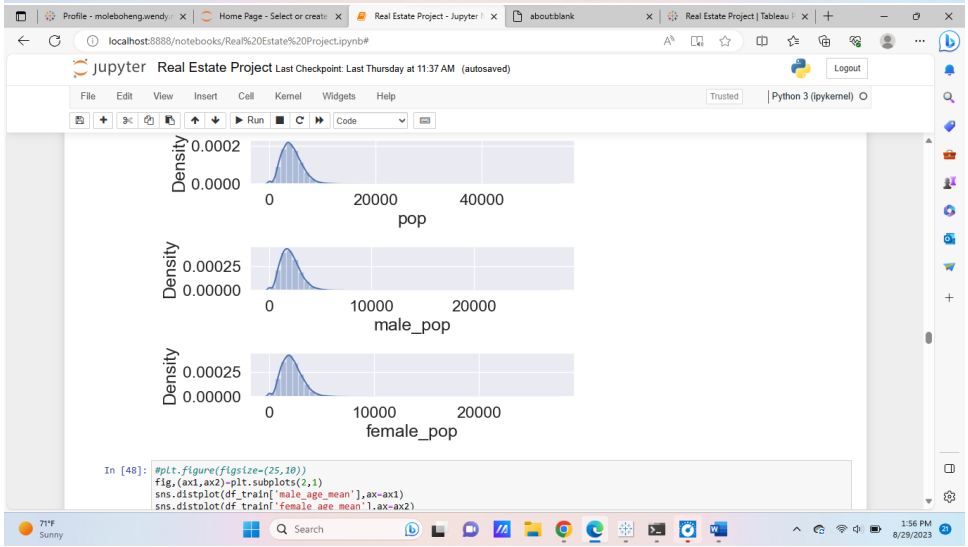
```
C:\Users\Student_0002\AppData\Local\Temp\ipykernel_11988\3130637729.py:1: UserWarning:


`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
```

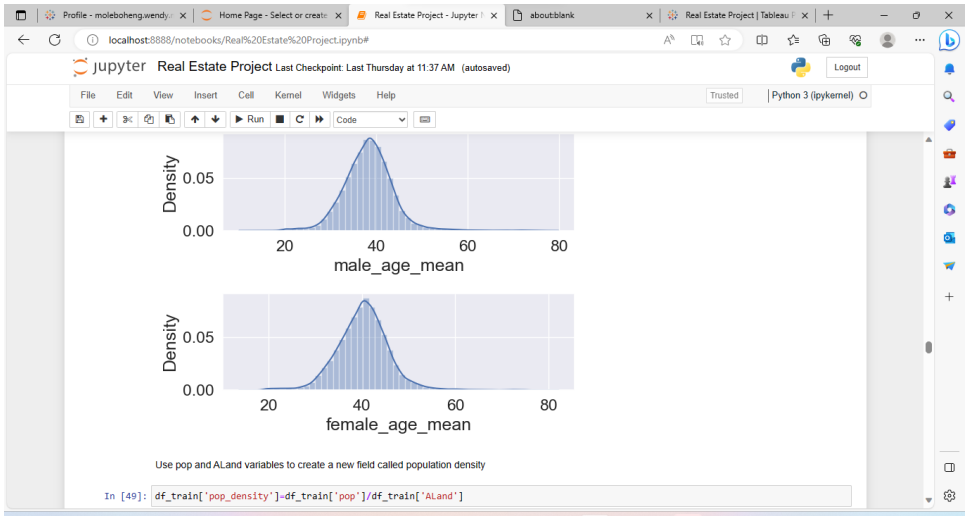




```
In [46]: sns.distplot(df_train['family_mean']-df_train['hi_mean'])
         plt.title('Remaining income distribution chart')
         plt.show()
```


Remaining income distribution chart
1e−5
Density
−200000
−100000
0
100000
200000
Income distribution almost has normality in its distrbution

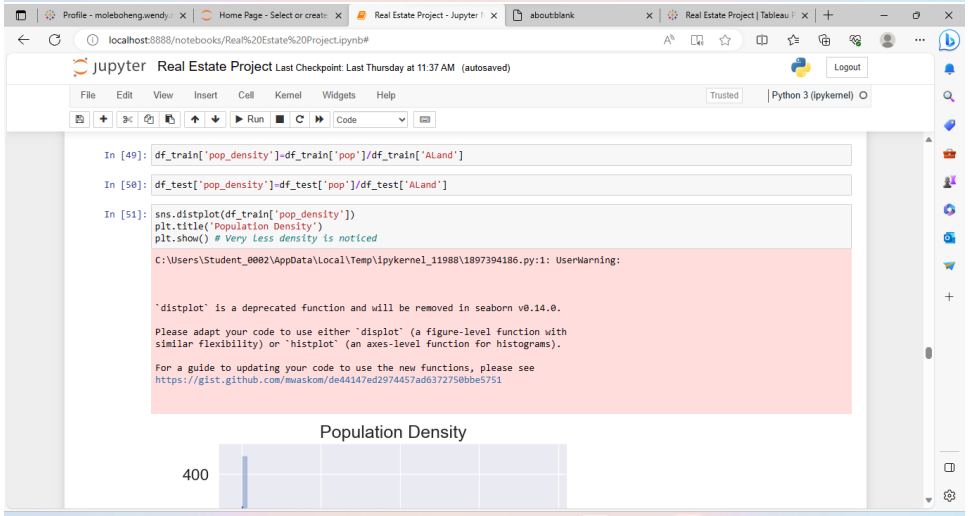

In [47]:
#plt.figure(figsize=(25,10))
fig,(ax1,ax2,ax3)=plt.subplots(3,1)
sns.distplot(df_train['pop'],ax=ax1)
sns.distplot(df_train['male_pop'],ax=ax2)
sns.distplot(df_train['female_pop'],ax=ax3)
plt.subplots_adjust(wspace=0.8,hspace=0.8)
plt.tight_layout()
plt.show()
C:\Users\Student_0002\AppData\Local\Temp\ipykernel_11988\222623768.py:3: UserWarning:
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).
For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
C:\Users\Student_0002\AppData\Local\Temp\ipykernel_11988\222623768.py:4: UserWarning:
`distplot` is a deprecated function and will be removed in seaborn v0.14.0.
Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).


Density
pop
male_pop
female_pop
In [48]:
#plt.figure(figsize=(25,10))
fig,(ax1,ax2)=plt.subplots(2,1)
sns.distplot(df_train['male_age_mean'],ax=ax1)
sns.distplot(df_train['female_age_mean'],ax=ax2)



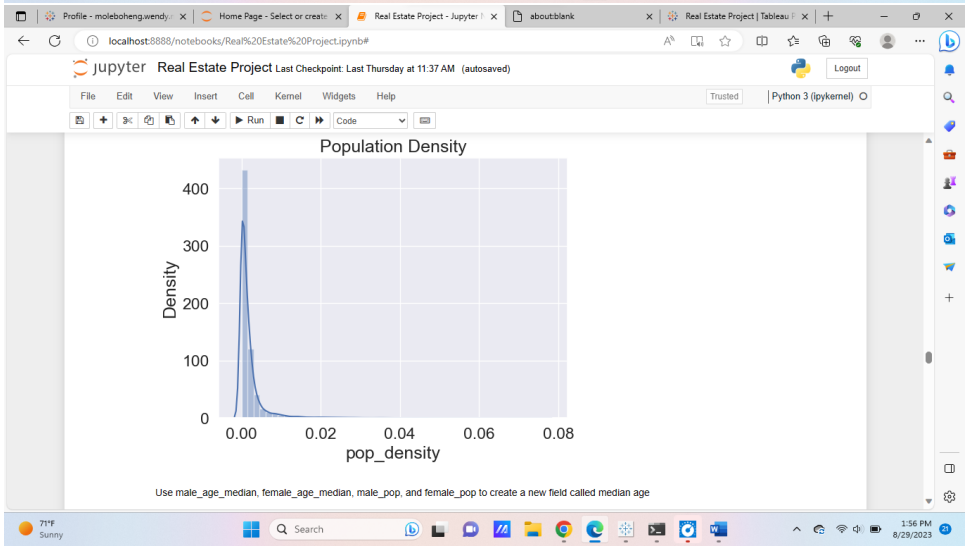
Use pop and ALand variables to create a new field called population density

```
In [49]: df_train['pop_density']=df_train['pop']/df_train['ALand']
```



```
In [49]: df_train['pop_density']=df_train['pop']/df_train['ALand']

In [50]: df_test['pop_density']=df_test['pop']/df_test['ALand']

In [51]: sns.distplot(df_train['pop_density'])
         plt.title('Population Density')
         plt.show() # Very Less density is noticed
```

```
C:\Users\Student_0002\AppData\Local\Temp\ipykernel_11988\1897394186.py:1: UserWarning:


`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751
```

Use male_age_median, female_age_median, male_pop, and female_pop to create a new field called median age

jupyter Real Estate Project Last Checkpoint: Last Thursday at 11:37 AM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

In [52]:
```
df_train['age_median']=(df_train['male_age_median']+df_train['female_age_median'])/2
df_test['age_median']=(df_test['male_age_median']+df_test['female_age_median'])/2
```

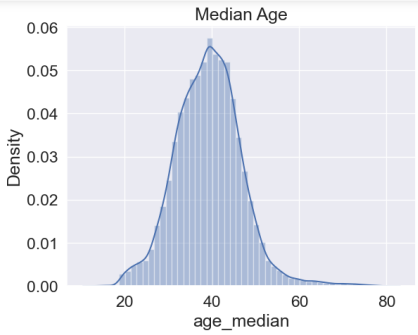
In [53]:
```
df_train[['male_age_median','female_age_median','male_pop','female_pop','age_median']].head()
```

Out[53]:

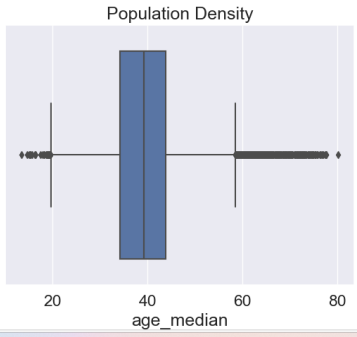
UID	male_age_median	female_age_median	male_pop	female_pop	age_median
267822	44.00000	45.33333	2612	2618	44.666665
246444	32.00000	37.58333	1349	1284	34.791665
245683	40.83333	42.83333	3643	3238	41.833330
279653	48.91667	50.58333	1141	1559	49.750000
247218	22.41667	21.58333	2586	3051	22.000000

In [54]:
```
sns.distplot(df_train['age_median'])
plt.title('Median Age')
plt.show()
# Age of population is mostly between 20 and 60
# Majority are of age around 40
# Median age distribution has a gaussian distribution
# Some right skewness is noticed
```

```
C:\Users\Student_0002\AppData\Local\Temp\ipykernel_11988\195219963.py:1: UserWarning:


`distplot` is a deprecated function and will be removed in seaborn v0.14.0.
```



In [55]:
```
sns.boxplot(data = df_train,x = 'age_median')
plt.title('Population Density')
plt.show()
```



Create bins for population into a new variable by selecting appropriate class interval so that the number of categories don't exceed 5 for the ease of analysis.

In [56]:
```
df_train['pop'].describe()
```

Out[56]:
```
count    27321.000000
mean      4316.032685
std       2169.226173
min          0.000000
25%       2885.000000
50%       4042.000000
75%       5430.000000
max      53812.000000
Name: pop, dtype: float64
```

In [57]:
```
df_train['pop_bins']=pd.cut(df_train['pop'],bins=5,labels=['very low','low','medium','high','very high'])
```

In [58]:
```
df_train[['pop','pop_bins']]
```

Out[58]:

UID	pop	pop_bins
267822	5230	very low
246444	2633	very low
245683	6881	very low
279653	2700	very low
247218	5637	very low
279212	1847	very low
277856	4155	very low
233000	2829	very low
287425	11542	low
265371	3726	very low

27321 rows × 2 columns

In [59]:
```
df_train['pop_bins'].value_counts()
```

Out[59]:
```
very low     27058
low            246
medium           9
high             7
very high        1
Name: pop_bins, dtype: int64
```

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

In [60]:
```
df_train.groupby(by='pop_bins')[['married','separated','divorced']].count()
```

Out[60]:

pop_bins	married	separated	divorced
very low	27058	27058	27058
low	246	246	246
medium	9	9	9
high	7	7	7
very high	1	1	1

In [61]:
```
df_train.groupby(by='pop_bins')[['married','separated','divorced']].agg(["mean", "median"])
```

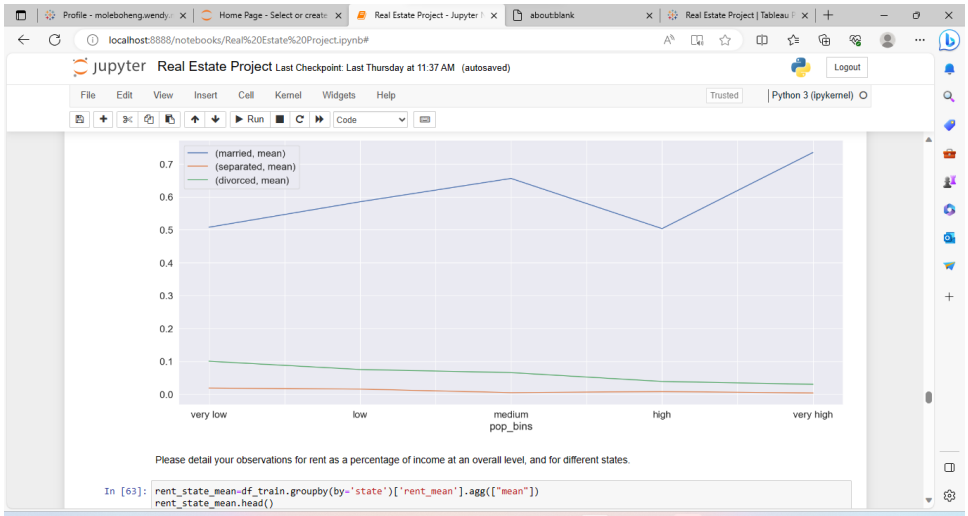
Out[61]:

pop_bins	married		separated		divorced	
	mean	median	mean	median	mean	median
very low	0.507548	0.524680	0.019126	0.013650	0.100504	0.096020
low	0.584894	0.593135	0.015833	0.011195	0.075348	0.070045
medium	0.655737	0.618710	0.005003	0.004120	0.065927	0.064890
high	0.503359	0.335660	0.008141	0.002500	0.039030	0.010320
very high	0.734740	0.734740	0.004050	0.004050	0.030360	0.030360

Visualize using appropriate chart type

In [62]:
```
plt.figure(figsize=(10,5))
pop_bin_married=df_train.groupby(by='pop_bins')[['married','separated','divorced']].agg(["mean"])
pop_bin_married.plot(figsize=(20,8))
plt.legend(loc='best')
plt.show()
```

```
<Figure size 1000x500 with 0 Axes>
```



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

In [63]:
```
rent_state_mean=df_train.groupby(by='state')['rent_mean'].agg(["mean"])
rent_state_mean.head()
```

Out[63]:

state	mean
Alabama	774.004927
Alaska	1185.763570
Arizona	1097.753511
Arkansas	720.918575
California	1471.133857

In [64]:
```
income_state_mean=df_train.groupby(by='state')['family_mean'].agg(["mean"])
income_state_mean.head()
```

Out[64]:

state	mean
Alabama	67030.064213
Alaska	92136.545109
Arizona	73328.238798
Arkansas	64765.377850
California	87655.470820

In [65]:
```
rent_perc_of_income=rent_state_mean['mean']/income_state_mean['mean']
rent_perc_of_income.head(10)
```

Out[65]:
```
state
Alabama                 0.011547
Alaska                  0.012870
Arizona                 0.014970
Arkansas                0.011131
California              0.016783
Colorado                0.013529
Connecticut             0.012637
Delaware                0.012929
District of Columbia    0.013198
Florida                 0.015772
Name: mean, dtype: float64
```

In [66]:
```
#overall level rent as a percentage of income
sum(df_train['rent_mean'])/sum(df_train['family_mean'])
```

Out[66]: 0.013358170721473864

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

In [67]:
```
df_train.columns
```

Out[67]:
```
Index(['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',
```

Profile - moleboheng.wendy | Home Page - Select or create | Real Estate Project - Jupyter | about:blank | Real Estate Project | Tableau

localhost:8888/notebooks/Real%20Estate%20Project.ipynb#

jupyter **Real Estate Project** Last Checkpoint: Last Thursday at 11:37 AM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [67]:
```
df_train.columns
```

Out[67]:
```
Index(['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', 'bins', 'pop_density', 'age_median', 'pop_bins'],
      dtype='object')
```

In [68]:
```
cor=df_train[['COUNTYID','STATEID','zip_code','type','pop', 'family_mean',
              'second_mortgage', 'home_equity', 'debt','hs_degree',
              'age_median','pct_own', 'married','separated', 'divorced']].corr()
```

```
C:\Users\Student_0002\AppData\Local\Temp\ipykernel_11988\3214557709.py:3: FutureWarning:

The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only v
```

71°F Sunny | Search | 1:58 PM 8/29/2023

Profile - moleboheng.wendy | Home Page - Select or create | Real Estate Project - Jupyter | about:blank | Real Estate Project | Tableau

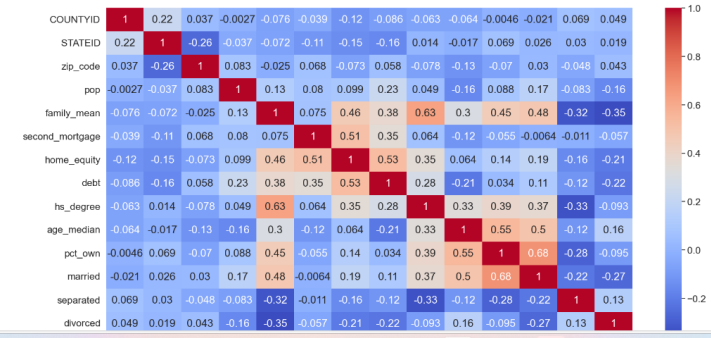
localhost:8888/notebooks/Real%20Estate%20Project.ipynb#

jupyter **Real Estate Project** Last Checkpoint: Last Thursday at 11:37 AM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code

In [69]:
```
plt.figure(figsize=(20,10))
sns.heatmap(cor,annot=True,cmap='coolwarm')
plt.show()
```



71°F Sunny | Search | 1:58 PM 8/29/2023

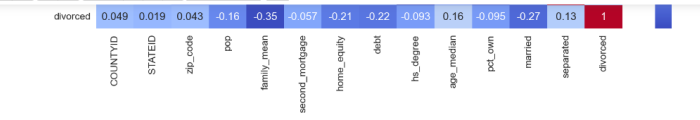
Profile - moleboheng.wendy | Home Page - Select or create | Real Estate Project - Jupyter | about:blank | Real Estate Project | Tableau

localhost:8888/notebooks/Real%20Estate%20Project.ipynb#

jupyter **Real Estate Project** Last Checkpoint: Last Thursday at 11:37 AM (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

Run Code



High positive correaltion is noticed between pop, male_pop and female_pop

High positive correaltion is noticed between rent_mean,hi_mean, family_mean,hc_mean

Project Task: Week 2

Data Pre-processing:

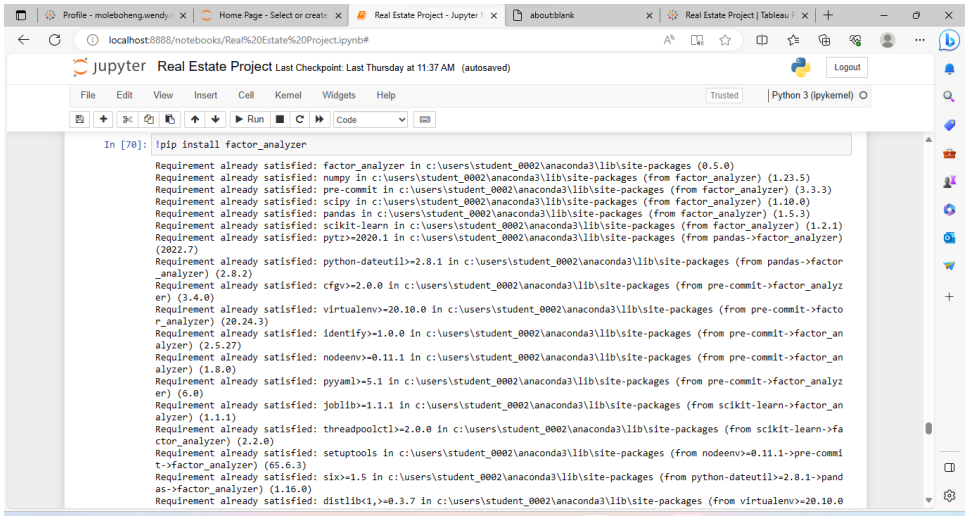
The economic multivariate data has a significant number of measured variables. The goal is to find where the measured variables depend on a number of smaller unobserved common factors or latent variables.

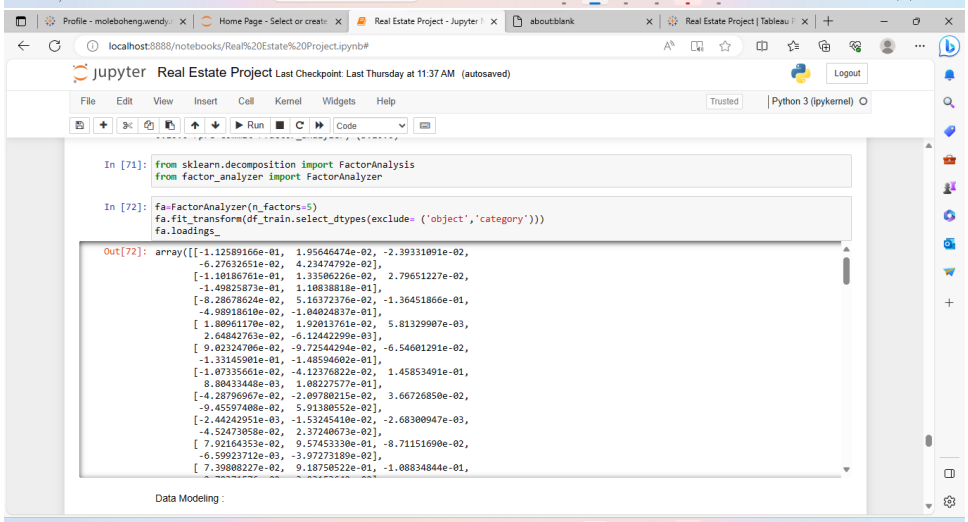
Each variable is assumed to be dependent upon a linear combination of the common factors, and the coefficients are known as loadings. Each measured variable also includes a component due to independent random variability, known as "specific variance" because it is specific to one variable. Obtain the common factors and then plot the loadings. Use factor analysis to find latent variables in our dataset and gain insight into the linear relationships in the data.

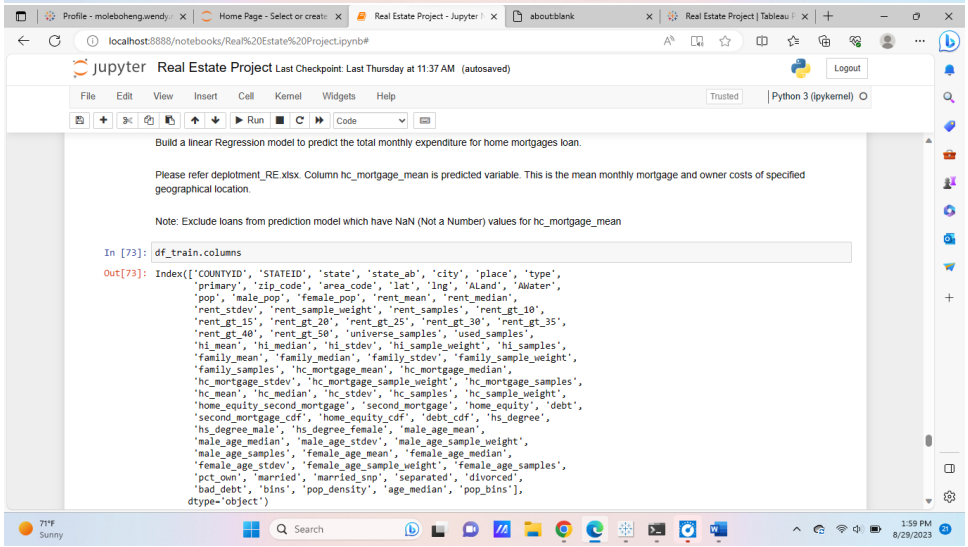
In [70]:
```
!pip install factor_analyzer
```

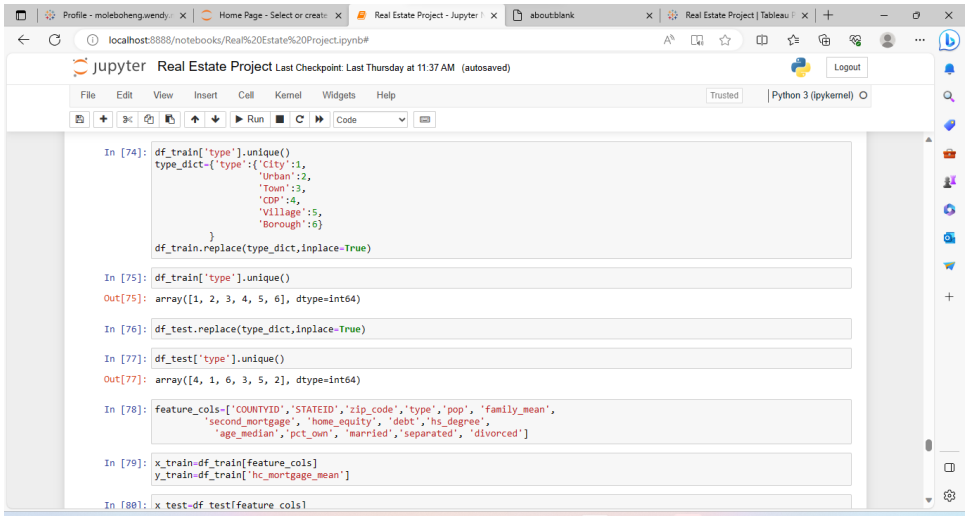
```
Requirement already satisfied: factor_analyzer in c:\users\student_0002\anaconda3\lib\site-packages (0.5.0)
```

71°F Sunny | Search | 1:59 PM 8/29/2023


jupyter Real Estate Project Last Checkpoint: Last Thursday at 11:37 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [70]: !pip install factor_analyzer
Requirement already satisfied: factor_analyzer in c:\users\student_0002\anaconda3\lib\site-packages (0.5.0)
Requirement already satisfied: numpy in c:\users\student_0002\anaconda3\lib\site-packages (from factor_analyzer) (1.23.5)
Requirement already satisfied: pre-commit in c:\users\student_0002\anaconda3\lib\site-packages (from factor_analyzer) (3.3.3)
Requirement already satisfied: scipy in c:\users\student_0002\anaconda3\lib\site-packages (from factor_analyzer) (1.10.0)
Requirement already satisfied: pandas in c:\users\student_0002\anaconda3\lib\site-packages (from factor_analyzer) (1.5.3)
Requirement already satisfied: scikit-learn in c:\users\student_0002\anaconda3\lib\site-packages (from factor_analyzer) (1.2.1)
Requirement already satisfied: pytz>=2020.1 in c:\users\student_0002\anaconda3\lib\site-packages (from pandas->factor_analyzer)
(2022.7)
Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\student_0002\anaconda3\lib\site-packages (from pandas->factor
_analyzer) (2.8.2)
Requirement already satisfied: cfgv>=2.0.0 in c:\users\student_0002\anaconda3\lib\site-packages (from pre-commit->factor_analyz
er) (3.4.0)
Requirement already satisfied: virtualenv>=20.10.0 in c:\users\student_0002\anaconda3\lib\site-packages (from pre-commit->facto
r_analyzer) (20.24.3)
Requirement already satisfied: identify>=1.0.0 in c:\users\student_0002\anaconda3\lib\site-packages (from pre-commit->factor_an
alyzer) (2.5.27)
Requirement already satisfied: nodeenv>=0.11.1 in c:\users\student_0002\anaconda3\lib\site-packages (from pre-commit->factor_an
alyzer) (1.8.0)
Requirement already satisfied: pyyaml>=5.1 in c:\users\student_0002\anaconda3\lib\site-packages (from pre-commit->factor_analyz
er) (6.0)
Requirement already satisfied: joblib>=1.1.1 in c:\users\student_0002\anaconda3\lib\site-packages (from scikit-learn->factor_an
alyzer) (1.1.1)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\student_0002\anaconda3\lib\site-packages (from scikit-learn->fa
ctor_analyzer) (2.2.0)
Requirement already satisfied: setuptools in c:\users\student_0002\anaconda3\lib\site-packages (from nodeenv>=0.11.1->pre-commi
t->factor_analyzer) (65.6.3)
Requirement already satisfied: six>=1.5 in c:\users\student_0002\anaconda3\lib\site-packages (from python-dateutil>=2.8.1->pand
as->factor_analyzer) (1.16.0)
Requirement already satisfied: distlib<1,>=0.3.7 in c:\users\student_0002\anaconda3\lib\site-packages (from virtualenv>=20.10.0
71°F Sunny
Search
1:59 PM 8/29/2023


jupyter Real Estate Project Last Checkpoint: Last Thursday at 11:37 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
In [71]: from sklearn.decomposition import FactorAnalysis
from factor_analyzer import FactorAnalyzer
In [72]: fa=FactorAnalyzer(n_factors=5)
fa.fit_transform(df_train.select_dtypes(exclude= ('object','category')))
fa.loadings_
Out[72]: array([[-1.12589166e-01, 1.95646474e-02, -2.39331091e-02,
-6.27632651e-02, 4.23474792e-02],
[-1.10186761e-01, 1.33506226e-02, 2.79651227e-02,
-1.49825873e-01, 1.10838818e-01],
[-8.28678624e-02, 5.16372376e-02, -1.36451866e-01,
-4.98918610e-02, -1.04024837e-01],
[1.80961170e-02, 1.92013761e-02, 5.81329907e-03,
2.64842763e-02, -6.12442299e-03],
[9.02324706e-02, -9.72544294e-02, -6.54601291e-02,
-1.33145901e-01, -1.48594602e-01],
[-1.07335661e-02, -4.12376822e-02, 1.45853491e-01,
8.80433448e-03, 1.08227577e-01],
[-4.28796967e-02, -2.09780215e-02, 3.66726850e-02,
-9.45597408e-02, 5.91308552e-02],
[-2.44242951e-03, -1.53245410e-02, -2.68300947e-03,
-4.52473058e-02, 2.37240673e-02],
[7.92164353e-02, 9.57453330e-01, -8.71151690e-02,
-6.59923712e-03, -3.97273189e-02],
[7.39808227e-02, 9.18750522e-01, -1.08834844e-01,
Data Modeling :
71°F Sunny
Search
1:59 PM 8/29/2023

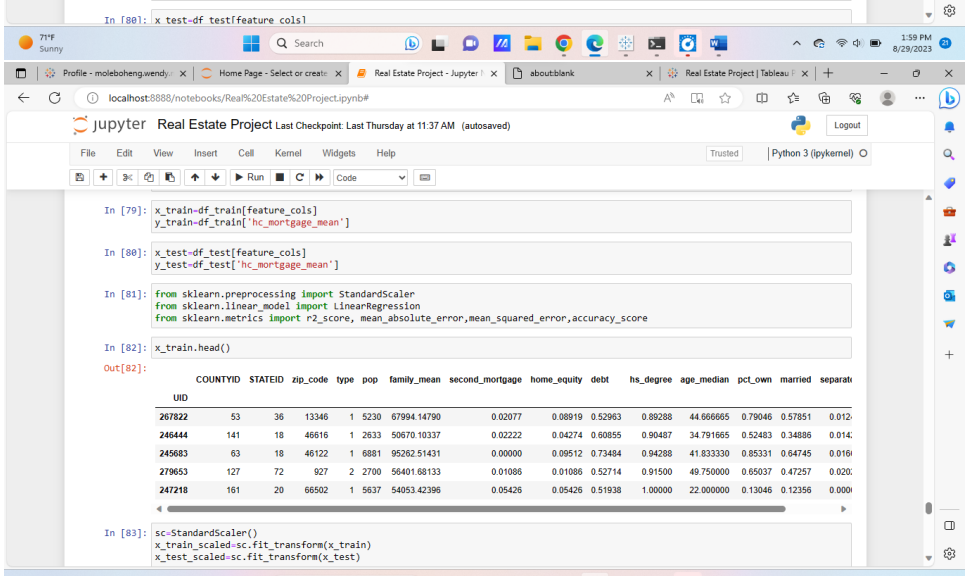

jupyter Real Estate Project Last Checkpoint: Last Thursday at 11:37 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Build a linear Regression model to predict the total monthly expenditure for home mortgages loan.
Please refer deplotment_RE.xlsx. Column hc_mortgage_mean is predicted variable. This is the mean monthly mortgage and owner costs of specified geographical location.
Note: Exclude loans from prediction model which have NaN (Not a Number) values for hc_mortgage_mean
In [73]: df_train.columns
Out[73]: Index(['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', 'bins', 'pop_density', 'age_median', 'pop_bins'],
dtype='object')
71°F Sunny
Search
1:59 PM 8/29/2023



jupyter Real Estate Project Last Checkpoint: Last Thursday at 11:37 AM (autosaved)

File Edit View Insert Cell Kernel Widgets Help

```
In [74]: df_train['type'].unique()
         type_dict={'type':{'City':1,
                            'Urban':2,
                            'Town':3,
                            'CDP':4,
                            'Village':5,
                            'Borough':6}
                    }
         df_train.replace(type_dict,inplace=True)

In [75]: df_train['type'].unique()

Out[75]: array([1, 2, 3, 4, 5, 6], dtype=int64)

In [76]: df_test.replace(type_dict,inplace=True)

In [77]: df_test['type'].unique()

Out[77]: array([4, 1, 6, 3, 5, 2], dtype=int64)

In [78]: feature_cols=['COUNTYID','STATEID','zip_code','type','pop', 'family_mean',
                       'second_mortgage', 'home_equity', 'debt','hs_degree',
                        'age_median','pct_own', 'married','separated', 'divorced']

In [79]: x_train=df_train[feature_cols]
         y_train=df_train['hc_mortgage_mean']

In [80]: x_test=df_test[feature_cols]
```



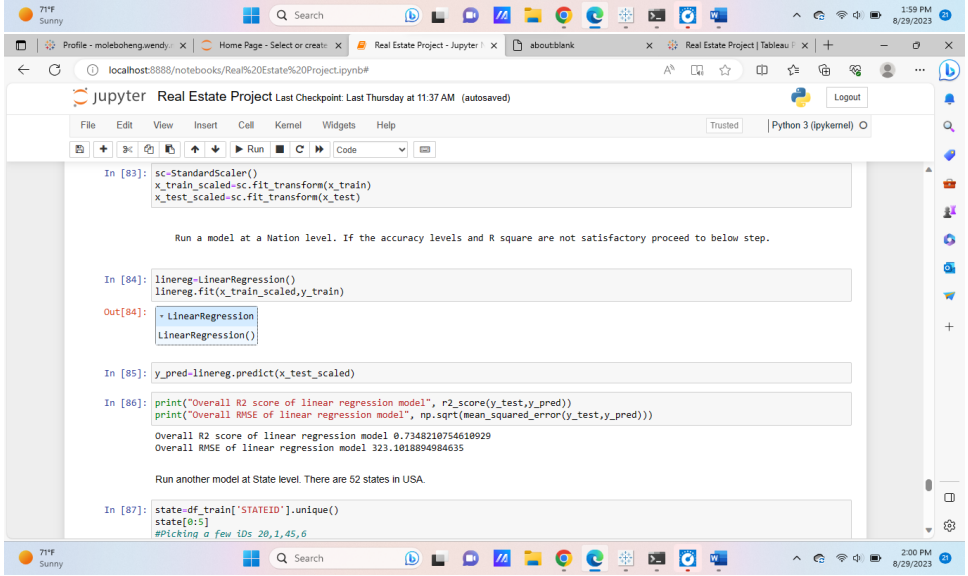
jupyter Real Estate Project Last Checkpoint: Last Thursday at 11:37 AM (autosaved)

File Edit View Insert Cell Kernel Widgets Help

```
In [79]: x_train=df_train[feature_cols]
         y_train=df_train['hc_mortgage_mean']

In [80]: x_test=df_test[feature_cols]
         y_test=df_test['hc_mortgage_mean']

In [81]: from sklearn.preprocessing import StandardScaler
         from sklearn.linear_model import LinearRegression
         from sklearn.metrics import r2_score, mean_absolute_error,mean_squared_error,accuracy_score

In [82]: x_train.head()

Out[82]:
```

UID	COUNTYID	STATEID	zip_code	type	pop	family_mean	second_mortgage	home_equity	debt	hs_degree	age_median	pct_own	married	separate
267822	53	36	13346	1	5230	67994.14790	0.02077	0.08919	0.52963	0.89288	44.666665	0.79046	0.57851	0.012
246444	141	18	46616	1	2633	50670.10337	0.02222	0.04274	0.60855	0.90487	34.791665	0.52483	0.34886	0.014
245683	63	18	46122	1	6881	95262.51431	0.00000	0.09512	0.73484	0.94288	41.833330	0.85331	0.64745	0.016
279653	127	72	927	2	2700	56401.68133	0.01086	0.01086	0.52714	0.91500	49.750000	0.65037	0.47257	0.020
247218	161	20	66502	1	5637	54053.42396	0.05426	0.05426	0.51938	1.00000	22.000000	0.13046	0.12356	0.000

```
In [83]: sc=StandardScaler()
         x_train_scaled=sc.fit_transform(x_train)
         x_test_scaled=sc.fit_transform(x_test)
```



jupyter Real Estate Project Last Checkpoint: Last Thursday at 11:37 AM (autosaved)

File Edit View Insert Cell Kernel Widgets Help

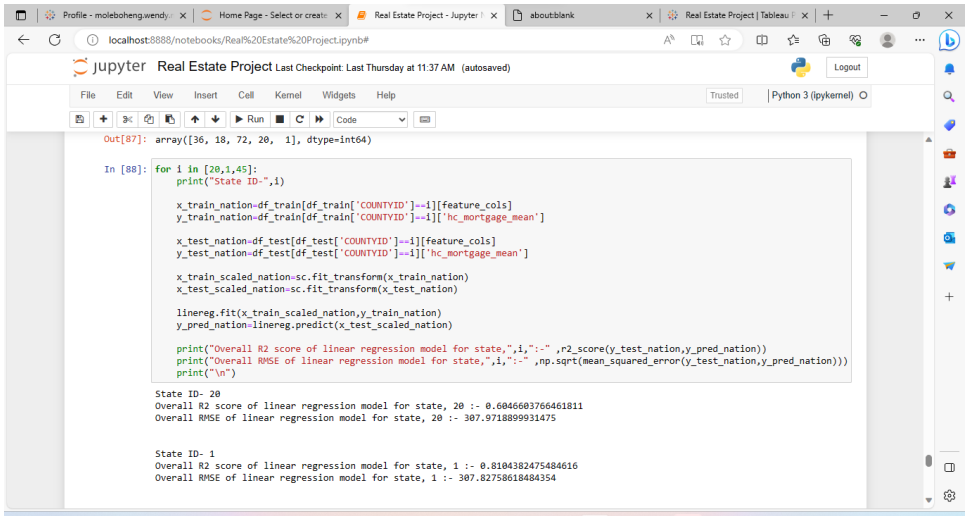
```
In [83]: sc=StandardScaler()
         x_train_scaled=sc.fit_transform(x_train)
         x_test_scaled=sc.fit_transform(x_test)
```

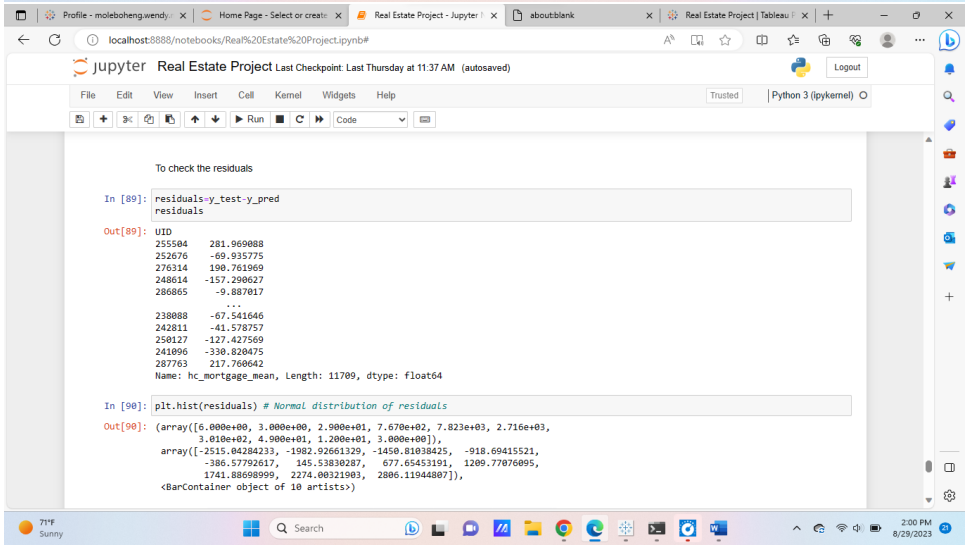
Run a model at a Nation level. If the accuracy levels and R square are not satisfactory proceed to below step.

```
In [84]: linereg=LinearRegression()
         linereg.fit(x_train_scaled,y_train)

Out[84]: LinearRegression
         LinearRegression()

In [85]: y_pred=linereg.predict(x_test_scaled)

In [86]: print("Overall R2 score of linear regression model", r2_score(y_test,y_pred))
         print("Overall RMSE of linear regression model", np.sqrt(mean_squared_error(y_test,y_pred)))

         Overall R2 score of linear regression model 0.7348210754610929
         Overall RMSE of linear regression model 323.1018894984635
```

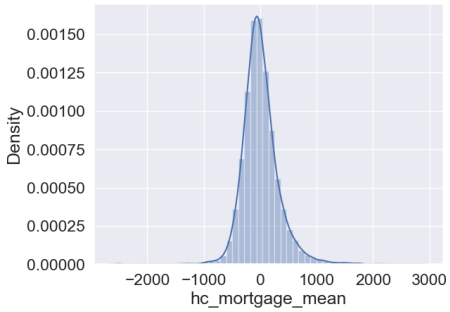
Run another model at State level. There are 52 states in USA.

```
In [87]: state=df_train['STATEID'].unique()
         state[0:5]
         #Picking a few ids 20,1,45,6
```


jupyter Real Estate Project Last Checkpoint: Last Thursday at 11:37 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Out[87]: array([36, 18, 72, 20, 1], dtype=int64)
In [88]:
for i in [20,1,45]:
print("State ID-",i)
x_train_nation=df_train[df_train['COUNTYID']==i][feature_cols]
y_train_nation=df_train[df_train['COUNTYID']==i]['hc_mortgage_mean']
x_test_nation=df_test[df_test['COUNTYID']==i][feature_cols]
y_test_nation=df_test[df_test['COUNTYID']==i]['hc_mortgage_mean']
x_train_scaled_nation=sc.fit_transform(x_train_nation)
x_test_scaled_nation=sc.fit_transform(x_test_nation)
linereg.fit(x_train_scaled_nation,y_train_nation)
y_pred_nation=linereg.predict(x_test_scaled_nation)
print("Overall R2 score of linear regression model for state,",i,":-" ,r2_score(y_test_nation,y_pred_nation))
print("Overall RMSE of linear regression model for state,",i,":-" ,np.sqrt(mean_squared_error(y_test_nation,y_pred_nation)))
print("\n")
State ID- 20
Overall R2 score of linear regression model for state, 20 :- 0.6046603766461811
Overall RMSE of linear regression model for state, 20 :- 307.9718899931475
State ID- 1
Overall R2 score of linear regression model for state, 1 :- 0.8104382475484616
Overall RMSE of linear regression model for state, 1 :- 307.82758618484354
71°F Sunny
Search
2:00 PM 8/29/2023


jupyter Real Estate Project Last Checkpoint: Last Thursday at 11:37 AM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
To check the residuals
In [89]:
residuals=y_test-y_pred
residuals
Out[89]: UID
255504 281.969088
252676 -69.935775
276314 190.761969
248614 -157.290627
286865 -9.887017
...
238088 -67.541646
242811 -41.578757
250127 -127.427569
241096 -330.820475
287763 217.760642
Name: hc_mortgage_mean, Length: 11709, dtype: float64
In [90]:
plt.hist(residuals) # Normal distribution of residuals
Out[90]: (array([6.000e+00, 3.000e+00, 2.900e+01, 7.670e+02, 7.823e+03, 2.716e+03,
3.010e+02, 4.900e+01, 1.200e+01, 3.000e+00]),
array([-2515.04284233, -1982.92661329, -1450.81038425, -918.69415521,
-386.57792617, 145.53830287, 677.65453191, 1209.77076095,
1741.88698999, 2274.00321903, 2806.11944807]),
<BarContainer object of 10 artists>)
71°F Sunny
Search
2:00 PM 8/29/2023

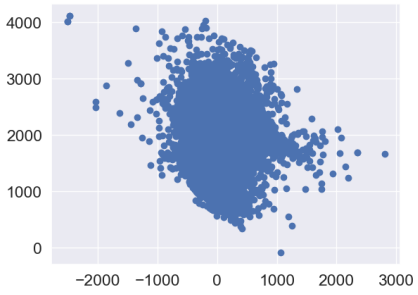
Out[91]: <Axes: xlabel='hc_mortgage_mean', ylabel='Density'>



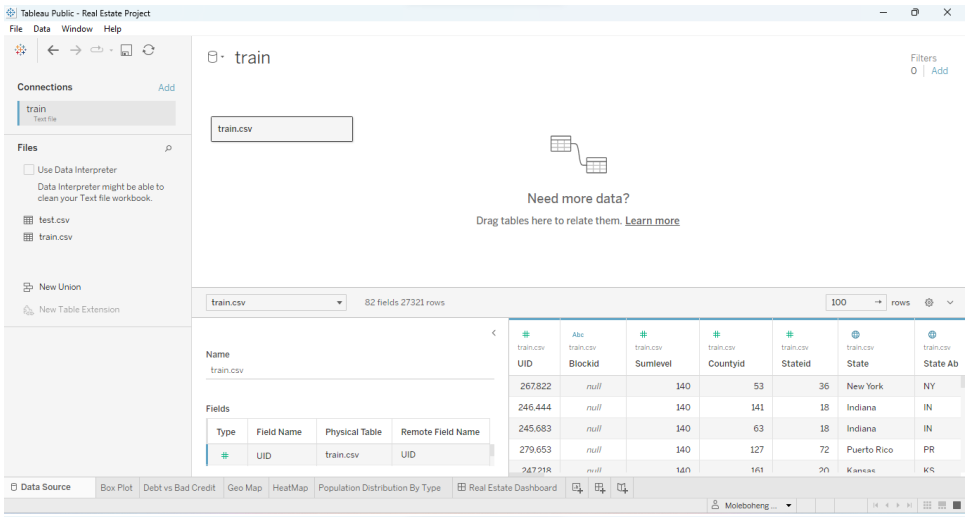
In [92]:
```
plt.scatter(residuals,y_pred) # Same variance and residuals does not have correlation with predictor
```

In [92]:
```
plt.scatter(residuals,y_pred) # Same variance and residuals does not have correlation with predictor
# Independance of residuals
```

Out[92]: <matplotlib.collections.PathCollection at 0x29414ea6cb0>



Screenshots Of Tableau Public Real Estate Project



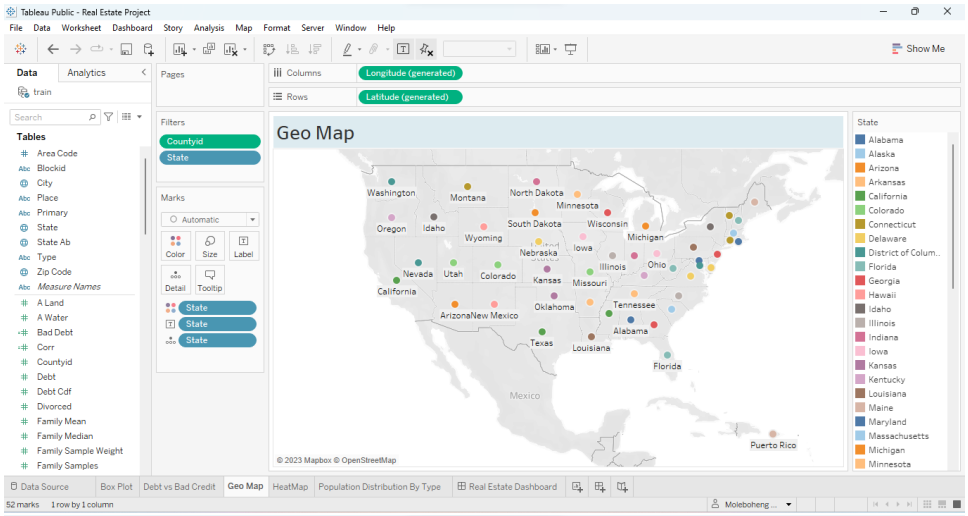


Tableau Public - Real Estate Project
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Show Me
Data
Analytics
train
Search
Tables
Area Code
Blockid
City
Place
Primary
State
State Ab
Type
Zip Code
Measure Names
A Land
A Water
Bad Debt
Corr
Countyid
Debt
Debt Cdf
Divorced
Family Mean
Family Median
Family Sample Weight
Family Samples
Pages
Filters
Countyid
State
Marks
Automatic
Color
Size
Label
Detail
Tooltip
State
State
State
Columns
Longitude (generated)
Rows
Latitude (generated)
Geo Map
Washington
Montana
North Dakota
Minnesota
Oregon
Idaho
South Dakota
Wisconsin
Wyoming
Michigan
Nebraska
Iowa
Illinois
Ohio
Nevada
Utah
Colorado
Kansas
Missouri
California
Oklahoma
Tennessee
ArizonaNew Mexico
Alabama
Texas
Louisiana
Florida
Mexico
Puerto Rico
© 2023 Mapbox © OpenStreetMap
State
Alabama
Alaska
Arizona
Arkansas
California
Colorado
Connecticut
Delaware
District of Colum..
Florida
Georgia
Hawaii
Idaho
Illinois
Indiana
Iowa
Kansas
Kentucky
Louisiana
Maine
Maryland
Massachusetts
Michigan
Minnesota
Data Source
Box Plot
Debt vs Bad Credit
Geo Map
HeatMap
Population Distribution By Type
Real Estate Dashboard
52 marks
1 row by 1 column
Moleboheng ...

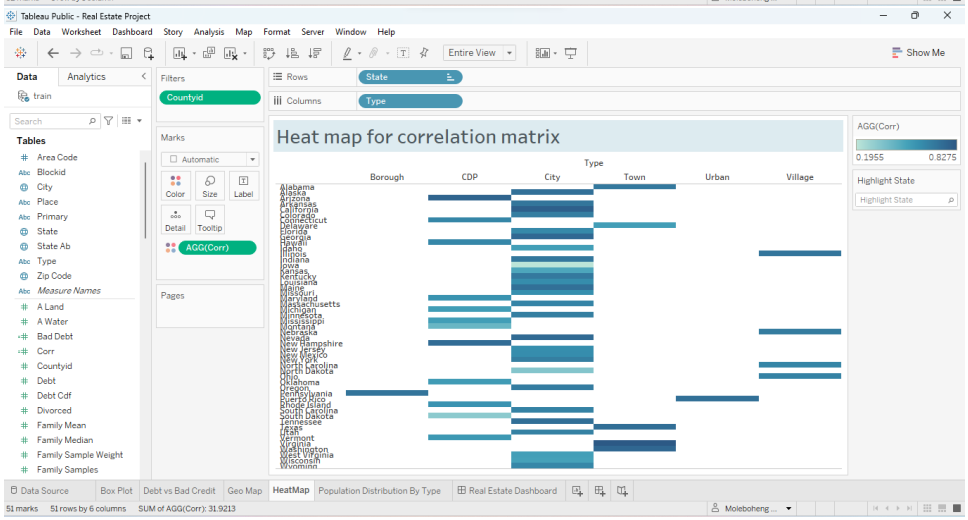
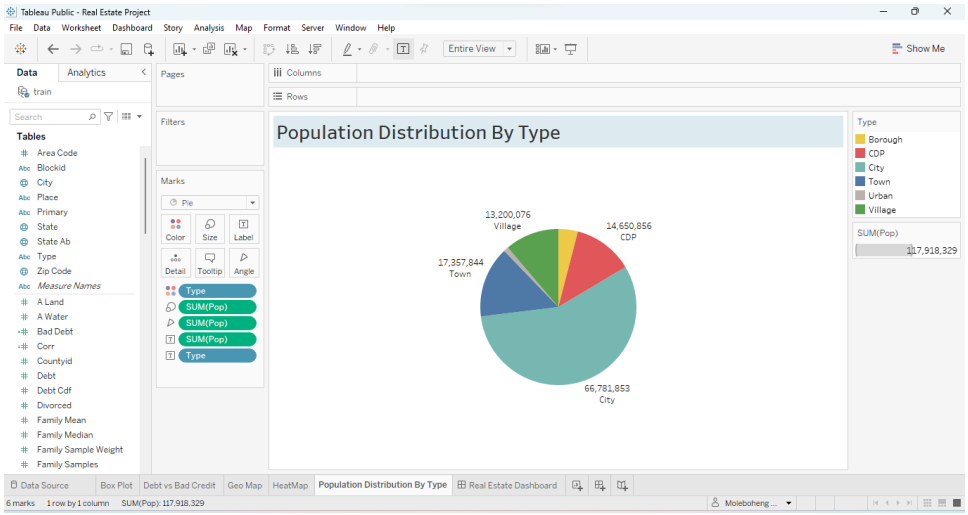
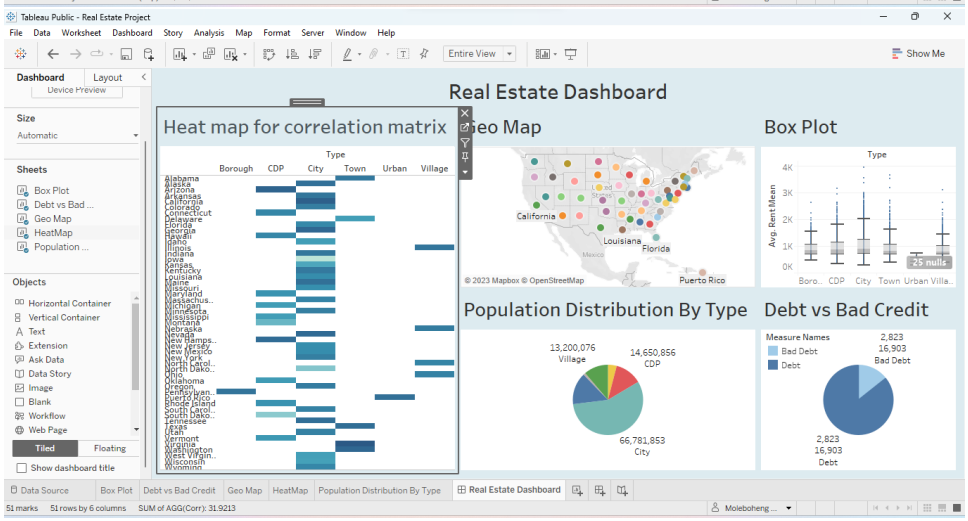


Tableau Public - Real Estate Project
File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Entire View
Show Me
Data
Analytics
train
Search
Tables
Area Code
Blockid
City
Place
Primary
State
State Ab
Type
Zip Code
Measure Names
A Land
A Water
Bad Debt
Corr
Countyid
Debt
Debt Cdf
Divorced
Family Mean
Family Median
Family Sample Weight
Family Samples
Pages
Filters
Countyid
Marks
Automatic
Color
Size
Label
Detail
Tooltip
AGG(Corr)
Rows
State
Columns
Type
Heat map for correlation matrix
Type
Borough
CDP
City
Town
Urban
Village
AGG(Corr)
0.1955
0.8275
Highlight State
Highlight State
Data Source
Box Plot
Debt vs Bad Credit
Geo Map
HeatMap
Population Distribution By Type
Real Estate Dashboard
51 marks
51 rows by 6 columns
SUM of AGG(Corr): 31.9213
Moleboheng ...





Link To Tableau Real Estate Project:

https://public.tableau.com/app/profile/moleboheng.wendy.mokoena/viz/RealEstateProject_16933092279690/BoxPlot