EDA Case Study:Houses

By Jacob Alamparambil

Overview

- Preprocessing stage
- mls
- Lot_acres
- Sqr_ft
- Hoa
- Year_built
- Kitchen features
- Garage null values
- Bathroom null values
- Taxes
- statistics

Preprocessing Stage

- Load csv file
- df.head()
- df.info()
- df.isnull().sum()

Remove duplicate records

1) Identify primary key : mls

```
df.iloc[0:,0]
```

	MLS
0	21530491
1	21529082
2	3054672
3	21919321
4	21306357
...	...
4995	21810382
4996	21908591
4997	21832452
4998	21900515
4999	4111490

Remove duplicate records

2) Display that column

```
df.iloc[0:,0]
```

	MLS
0	21530491
1	21529082
2	3054672
3	21919321
4	21306357
...	...
4995	21810382
4996	21908591
4997	21832452
4998	21900515
4999	4111490

5000 rows × 1 columns

dtype: int64

Remove duplicate records

3) Store the keys in a set and compare with actual column of keys

```
mls=set(df.iloc[0:,0])
```

```
len(mls)
```

```
5000
```

```
len(df.iloc[0:,0])
```

```
5000
```

Change lot_acres null values to median

Before

```
print(df.isnull().sum())
```

```
MLS                0
sold_price         0
zipcode            0
longitude          0
latitude           0
lot_acres         10
taxes              0
year_built         0
bedrooms           0
bathrooms          6
sqrt_ft           56
garage             7
kitchen_features   33
fireplaces         0
floor_covering     1
HOA               562
```

lot_acres

0.99

0.99

0.99

0.99

After

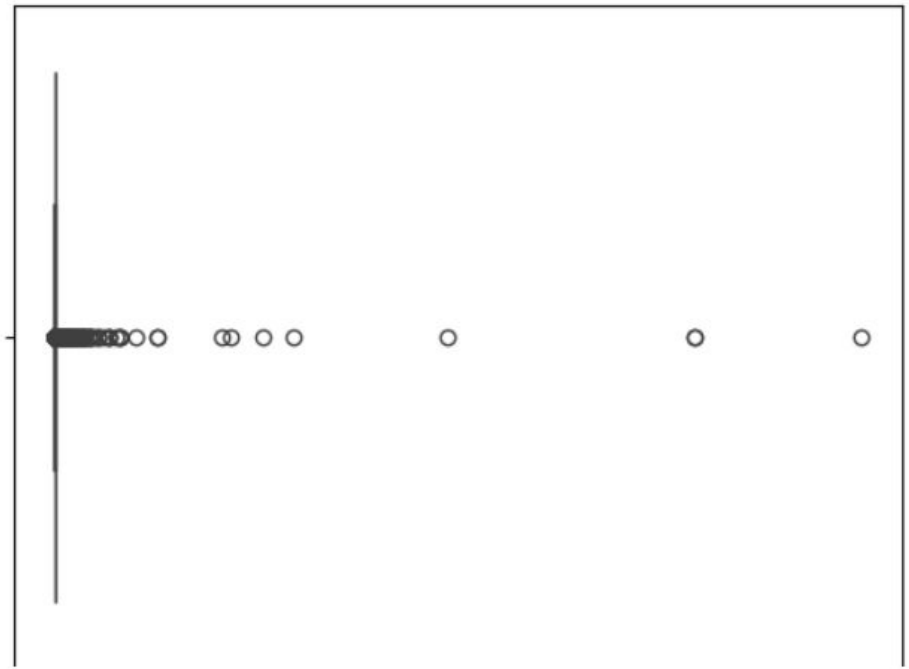
```
print(df.isnull().sum())
```

```
MLS                0
sold_price         0
zipcode            0
longitude          0
latitude           0
lot_acres          0
taxes              0
year_built         0
bedrooms           0
bathrooms          6
sqrt_ft           56
garage             7
kitchen_features   33
fireplaces         0
floor_covering     1
HOA               562
```

Lot_acres view

```
sns.boxplot(x=df['lot_acres'])
```

```
<Axes: xlabel='lot_acres'>
```



Why?

The data is skewed right to the lower values and contains many outliers so the appropriate imputation would be the median

Change sqr_ft null values to median

Why?

The data is skewed right to the lower values and contains many outliers so the appropriate imputation would be the median

Before

```
print(df.isnull().sum())
```

```
MLS                0
sold_price         0
zipcode            0
longitude          0
latitude           0
lot_acres          0
taxes              0
year_built         0
bedrooms           0
bathrooms          6
sqrt_ft           56
garage             7
kitchen_features   33
fireplaces         0
floor_covering     1
HOA              562
```

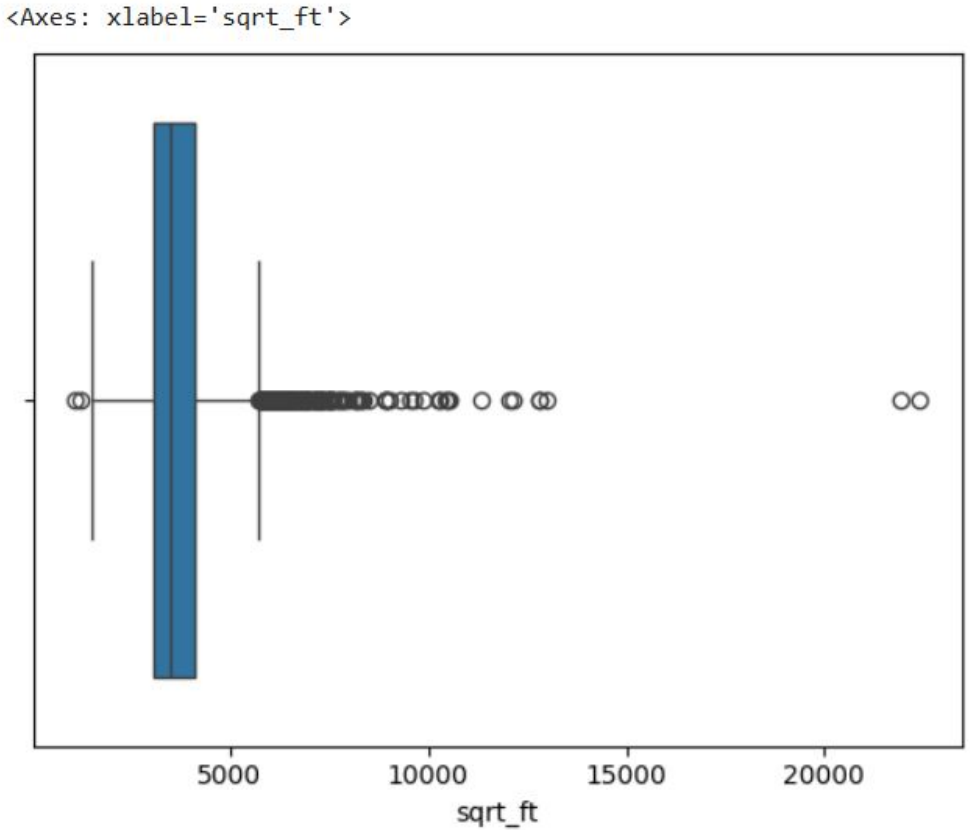
sqrt_ft
3512.0
3512.0
3512.0
3512.0

After

```
print(df.isnull().sum())
```

```
MLS                0
sold_price         0
zipcode            0
longitude          0
latitude           0
lot_acres          0
taxes              0
year_built         0
bedrooms           0
bathrooms          6
sqrt_ft            0
garage             7
kitchen_features   33
fireplaces         0
floor_covering     1
HOA              562
```

sqr_ft view



Why?

The data is skewed right to the lower values and contains many outliers so the appropriate imputation would be the median

Change hoa null value:None to median

Why is hoa=0 ok?

```
print(df.isnull().sum())
```

```
MLS                0
sold_price         0
zipcode            0
longitude          0
latitude           0
lot_acres          0
taxes              0
year_built         0
bedrooms           0
bathrooms          6
sqrt_ft            0
garage             7
kitchen_features   33
fireplaces         0
floor_covering     1
HOA                562
dtype: int64
```

```
print(df.isnull().sum())
```

```
MLS                0
sold_price         0
zipcode            0
longitude          0
latitude           0
lot_acres          0
taxes              0
year_built         0
bedrooms           0
bathrooms          6
sqrt_ft            0
garage             7
kitchen_features   33
fireplaces         0
floor_covering     1
HOA                0
dtype: int64
```

Change properties with $0 taxes to median value

```
df.iloc[100:,:]
```

	MLS	sold_price	zipcode	longitude	latitude	lot_acres	taxes	year_built	bedrooms	b
100	21522589	2100000.0	85718	-110.885186	32.327907	1.55	15668.99	2009	5	
101	21814436	1450000.0	85624	-110.738083	31.533643	164.30	0.00	1997	1	
102	21326674	1700000.0	85750	-110.846362	32.329630	1.12	17898.00	2002	4	

```
df.iloc[100:,:]
```

	MLS	sold_price	zipcode	longitude	latitude	lot_acres	taxes	year_built	bedrooms	b
100	21522589	2100000.0	85718	-110.885186	32.327907	1.55	15668.99	2009	5	
101	21814436	1450000.0	85624	-110.738083	31.533643	164.30	6223.76	1997	1	
102	21326674	1700000.0	85750	-110.846362	32.329630	1.12	17898.00	2002	4	

Change properties with $0 taxes to median value pt2

Why?

The data is skewed right to the lower values and contains many outliers so the appropriate imputation would be the median

Change year built values to median

```
years= [years_rep if x == 0 else x for x in list(df['year_built'])]
```

```
len(df[df['year_built']==years_rep])
```

```
169
```

```
df['year_built']=years
```

```
df['year_built']!=0
```

	year_built
0	False
1	False
2	False
3	False
4	False
...	...
4995	False
4996	False
4997	False
4998	False
4999	False

5000 rows × 1 columns

Change kitchen features null values to mode

```
print(df.isnull().sum())
```

```
MLS                0
sold_price         0
zipcode            0
longitude          0
latitude           0
lot_acres          0
taxes              0
year_built         0
bedrooms           0
bathrooms          0
sqrt_ft            0
garage             0
kitchen_features   0
fireplaces         0
floor_covering     1
HOA                0
dtype: int64
```

Change garage null values to median

```
print(df.isnull().sum())
```

```
MLS                0
sold_price         0
zipcode            0
longitude          0
latitude           0
lot_acres          0
taxes              0
year_built         0
bedrooms           0
bathrooms          6
sqrt_ft            0
garage             7
kitchen_features   33
fireplaces         0
floor_covering     1
HOA                0
dtype: int64
```

```
print(df.isnull().sum())
```

```
MLS                0
sold_price         0
zipcode            0
longitude          0
latitude           0
lot_acres          0
taxes              0
year_built         0
bedrooms           0
bathrooms          6
sqrt_ft            0
garage             0
kitchen_features   33
fireplaces         0
floor_covering     1
HOA                0
dtype: int64
```

Roundup garage values to nearest whole number

garage	garage
3.0	3
3.0	3
4.5	5
3.0	3

Change bathroom null values to median

```
print(df.isnull().sum())
```

```
MLS                0
sold_price         0
zipcode            0
longitude          0
latitude           0
lot_acres          0
taxes              0
year_built         0
bedrooms           0
bathrooms          6
sqrt_ft            0
garage             0
kitchen_features   33
fireplaces         0
floor_covering     1
HOA                0
dtype: int64
```

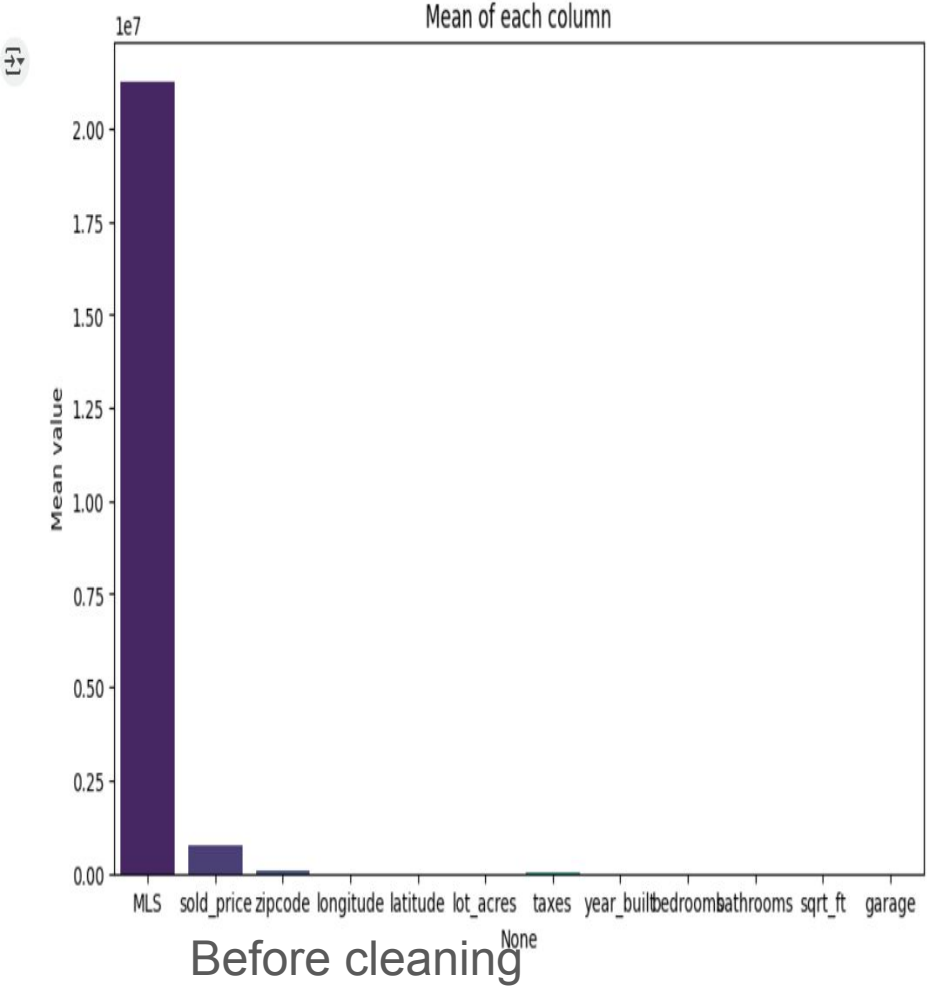
```
print(df.isnull().sum())
```

```
MLS                0
sold_price         0
zipcode            0
longitude          0
latitude           0
lot_acres          0
taxes              0
year_built         0
bedrooms           0
bathrooms          0
sqrt_ft            0
garage             7
kitchen_features   33
fireplaces         0
floor_covering     1
HOA                0
dtype: int64
```

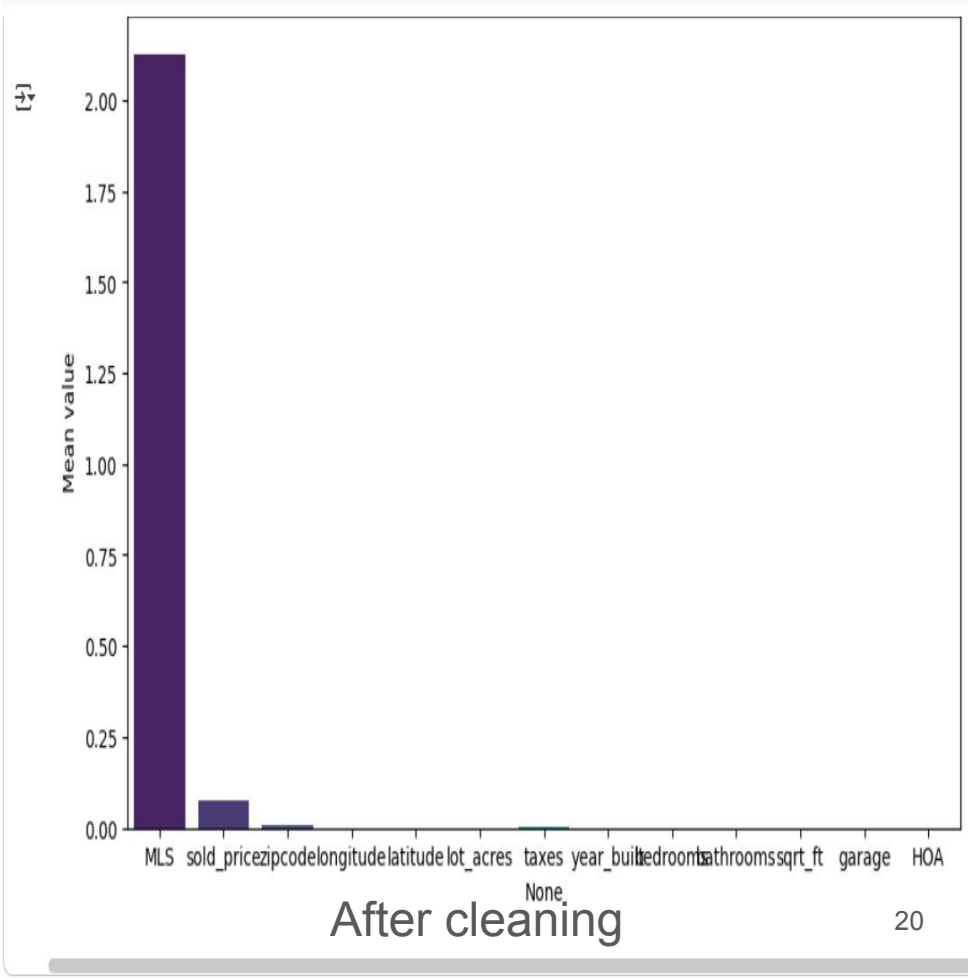
Comparisons of clean vs unclean data(mean)

index	mean(dirty)	mean(clean)	% Difference
MLS	21270699.22	21270699.22	0
sold_price	774626.2023	774626.2023	0
zipcode	85723.0256	85723.0256	0
longitude	-110.9121066	-110.9121066	0
latitude	32.30851202	32.30851202	0
lot_acres	4.661316633	4.653974	0.1575227254
taxes	9402.828094	9430.212638	0.2912373142
year_built	1992.328	1992.328	0
bedrooms	3.9338	3.9338	0
bathrooms	3.829895875	3.8301	0.005329779364
sqrt_ft	3716.366828	3714.07792	0.06158993732
garage	2.8161426	2.8232	0.2506052073

Comparisons of clean vs unclean data(mean pt2)



Before cleaning

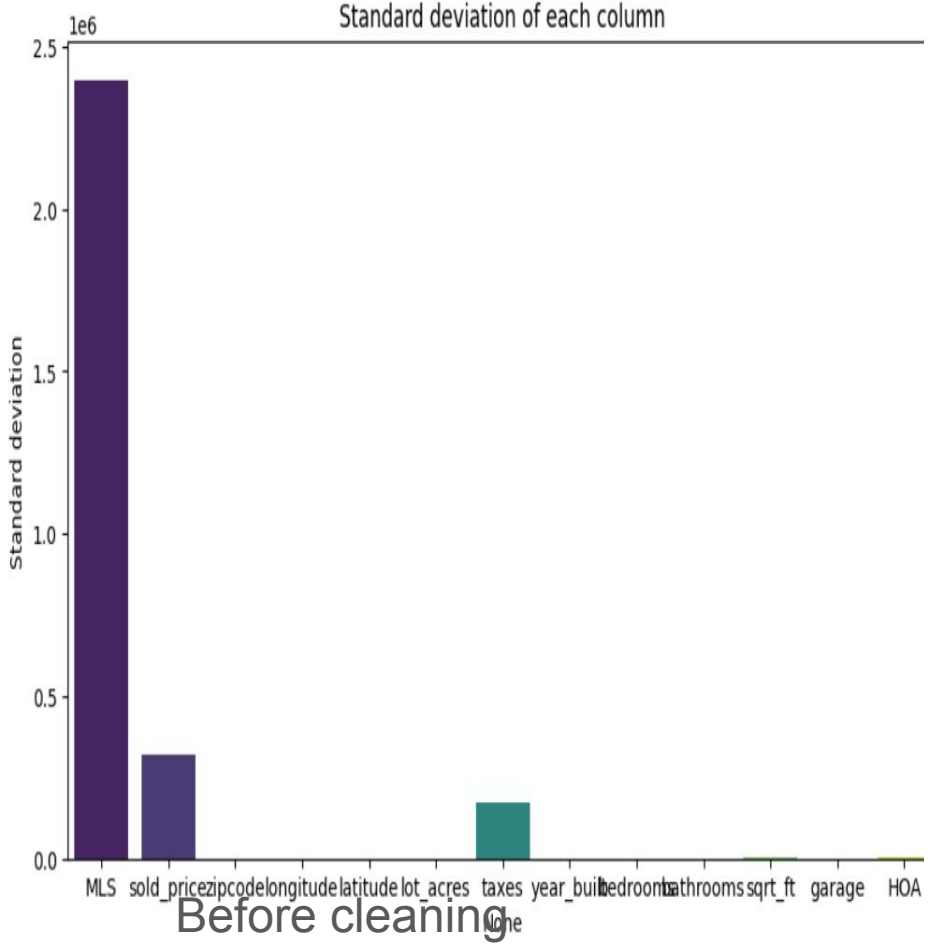


After cleaning

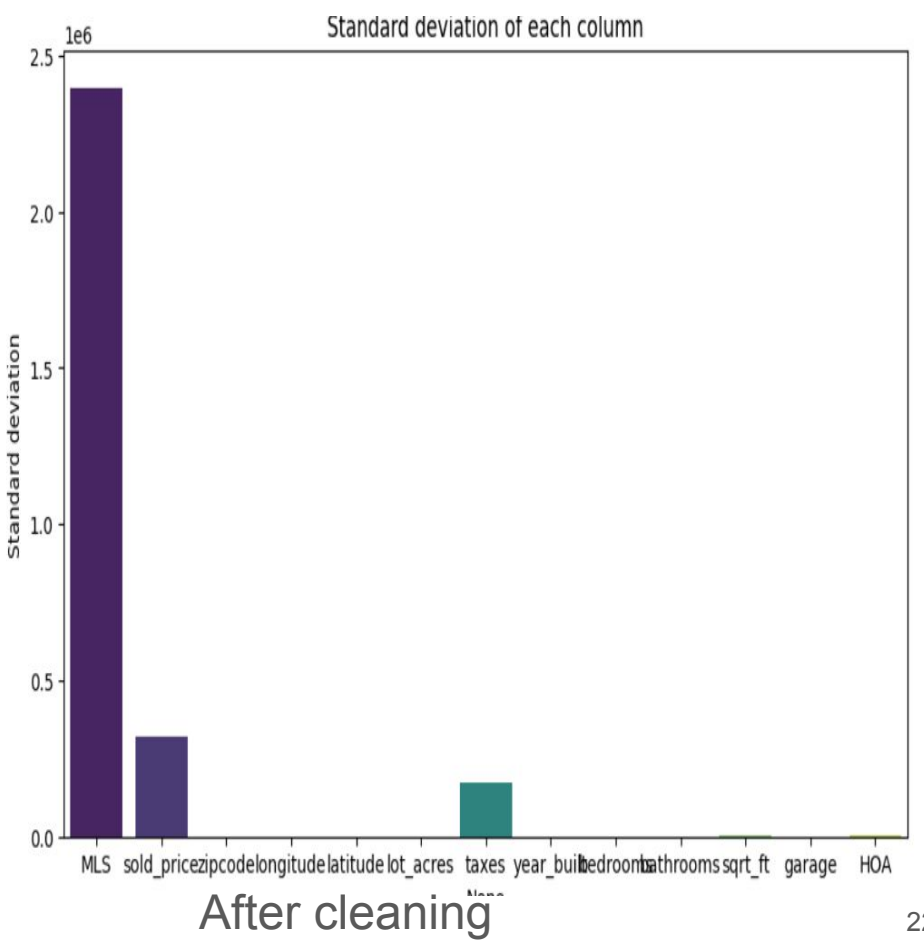
Comparisons of clean vs uncleaned data(std)

index	std(dirty)	std(clean)	% Difference
MLS	2.40E+06	2.40E+06	0.00E+00
sold_price	3.19E+05	3.19E+05	0.00E+00
zipcode	38.061712	38.061712	0
longitude	0.120629	0.120629	0
latitude	0.178028	0.178028	0
lot_acres	51.68523	51.633769	0.09956616233
taxes	1.73E+05	1.73E+05	5.78E-04
year_built	65.48614	65.48614	0
bedrooms	1.245362	1.245362	0
bathrooms	1.387063	1.386243	0.05911771852
sqrt_ft	1120.683515	1114.596305	0.5431694068
garage	1.192946	1.193745	0.06697704674
		average difference:	6.41E-02

Comparisons of clean vs uncleaned data(std pt2)



Before cleaning



After cleaning

Conclusion

- The number of rows stayed the same with a total of 5000 records
- Lot_acres,location,mls,,and sqr_ft are the most important columns because they determine the existence of the houses
- The data is very skewed and imputation has made it more normal distribution