import python libraries

In [98]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

read dataset in python file

In [170…
```
data=pd.read_csv('C:/Users/pv11379/Downloads/PropertyPrice_Data.csv')
```

In [100…
```
## view the dataset
data.head(2)
```

Out[100]:

	Id	Road_Type	Property_Shape	House_Type	House_Condition	Construction_Year	Remodel_Year	BsmtFinSF1	Total_Basement_Area	Air_Co
0	1	Paved	Reg	1Fam	5	2003	2003	706	856	
1	2	Paved	Reg	1Fam	8	1976	1976	978	1262	

2 rows × 26 columns

In [101…
```
## copy data into new variable
df=data.copy()
```

In [102…
```
## view last 2 rows of df
df.tail(2)
```

Out[102]:

	Id	Road_Type	Property_Shape	House_Type	House_Condition	Construction_Year	Remodel_Year	BsmtFinSF1	Total_Basement_Area
1457	1458	Paved	Reg	1Fam	9	1941	2006	275	1152
1458	1459	Paved	Reg	1Fam	6	1950	1996	49	1078

2 rows × 26 columns

In [103…
```
## shape of data
print('dataset is having total rows - ',df.shape[0])
print('dataset is having total columns - ',df.shape[1])
```

```
dataset is having total rows -  1459
dataset is having total columns -  26
```

In [104…
```
df.index
```

Out[104]:
```
RangeIndex(start=0, stop=1459, step=1)
```

In [105…
```
## data column names
coldf=df.columns
print(coldf)
```

```
Index(['Id', 'Road_Type', 'Property_Shape', 'House_Type', 'House_Condition',
       'Construction_Year', 'Remodel_Year', 'BsmtFinSF1',
       'Total_Basement_Area', 'Air_Conditioning', 'First_Floor_Area',
       'Second_Floor_Area', 'LowQualFinSF', 'Underground_Full_Bathroom',
       'Full_Bathroom_Above_Grade', 'Bedroom_Above_Grade', 'Kitchen_Quality',
       'Rooms_Above_Grade', 'Fireplaces', 'Garage', 'Garage_Built_Year',
       'Garage_Area', 'Pool_Area', 'Miscellaneous_Value', 'Year_Sold',
       'Sale_Price'],
      dtype='object')
```

In [106…
```
## change column names into lower case
col=[]
for i in coldf:
    a=i.lower()
    col.append(a)
```

In [107…
```
df.columns=col
```

In [108…
```
df.columns
```

Out[108]:
```
Index(['id', 'road_type', 'property_shape', 'house_type', 'house_condition',
       'construction_year', 'remodel_year', 'bsmtfinsf1',
       'total_basement_area', 'air_conditioning', 'first_floor_area',
       'second_floor_area', 'lowqualfinsf', 'underground_full_bathroom',
       'full_bathroom_above_grade', 'bedroom_above_grade', 'kitchen_quality',
       'rooms_above_grade', 'fireplaces', 'garage', 'garage_built_year',
       'garage_area', 'pool_area', 'miscellaneous_value', 'year_sold',
       'sale_price'],
      dtype='object')
```

In [70]:
```
## dataset detail information
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1459 entries, 0 to 1458
Data columns (total 26 columns):
 #   Column                      Non-Null Count  Dtype  
---  ------                      --------------  -----  
 0   id                          1459 non-null   int64  
 1   road_type                   1459 non-null   object 
 2   property_shape              1459 non-null   object 
 3   house_type                  1459 non-null   object 
 4   house_condition             1459 non-null   int64  
 5   construction_year           1459 non-null   int64  
 6   remodel_year                1459 non-null   int64  
 7   bsmtfinsf1                  1459 non-null   int64  
 8   total_basement_area         1459 non-null   int64  
 9   air_conditioning            1459 non-null   object 
 10  first_floor_area            1459 non-null   int64  
 11  second_floor_area           1459 non-null   int64  
 12  lowqualfinsf                1459 non-null   int64  
 13  underground_full_bathroom   1459 non-null   int64  
 14  full_bathroom_above_grade   1459 non-null   int64  
 15  bedroom_above_grade         1459 non-null   int64  
 16  kitchen_quality             1459 non-null   object 
 17  rooms_above_grade           1459 non-null   int64  
 18  fireplaces                  1459 non-null   int64  
 19  garage                      1378 non-null   object 
 20  garage_built_year           1378 non-null   float64
 21  garage_area                 1459 non-null   float64
 22  pool_area                   1459 non-null   int64  
 23  miscellaneous_value         1459 non-null   int64  
 24  year_sold                   1459 non-null   int64  
 25  sale_price                  1459 non-null   int64  
dtypes: float64(2), int64(18), object(6)
memory usage: 296.5+ KB
```

In [109…

```
## null values in columns are
df.isna().sum()
```

Out[109]:
```
id                          0
road_type                   0
property_shape              0
house_type                  0
house_condition             0
construction_year           0
remodel_year                0
bsmtfinsf1                  0
total_basement_area         0
air_conditioning            0
first_floor_area            0
second_floor_area           0
lowqualfinsf                0
underground_full_bathroom   0
full_bathroom_above_grade   0
bedroom_above_grade         0
kitchen_quality             0
rooms_above_grade           0
fireplaces                  0
garage                     81
garage_built_year          81
garage_area                 0
pool_area                   0
miscellaneous_value         0
year_sold                   0
sale_price                  0
dtype: int64
```

garage is object so replace null values with mode

In [110…
```
gar_count=df['garage'].value_counts()
gar_count
```

Out[110]:
```
Attchd     869
Detchd     387
BuiltIn     88
Basment     19
CarPort      9
2TFes        5
2Types       1
Name: garage, dtype: int64
```

In [111…
```
gar_mode=df['garage'].mode(0)
print(gar_mode[0])
```

```
Attchd
```

In [112…
```
df['garage'].fillna(gar_mode[0],inplace=True)
```

Garage_built_year is float, so replace null values with median

In [113…
```
df['garage_built_year'].head(2)
```

Out[113]:
```
0    2003.0
1    1976.0
Name: garage_built_year, dtype: float64
```

In [115…
```
gby_median=df['garage_built_year'].median(0)
gby_median
```

Out[115]:
```
1980.0
```

In [116…
```
df['garage_built_year'].fillna(gby_median,inplace=True)
```

In [117…
```
## check null values in garage_built_year and garage
df[['garage','garage_built_year']].isna().sum()
```

Out[117]:
```
garage              0
garage_built_year    0
dtype: int64
```

In [118…
```
## remove id column from dataset
df.drop('id',axis=1,inplace=True)
```

In [119…
```
df.shape
```

Out[119]:
```
(1459, 25)
```

check outliers in measure column and replace it.

In [120…

```
## Get measure column from dataset
ms_col=df.columns[df.dtypes!='O']
print(ms_col)
```

```
Index(['house_condition', 'construction_year', 'remodel_year', 'bsmtfinsf1',
       'total_basement_area', 'first_floor_area', 'second_floor_area',
       'lowqualfinsf', 'underground_full_bathroom',
       'full_bathroom_above_grade', 'bedroom_above_grade', 'rooms_above_grade',
       'fireplaces', 'garage_built_year', 'garage_area', 'pool_area',
       'miscellaneous_value', 'year_sold', 'sale_price'],
      dtype='object')
```

In [121…

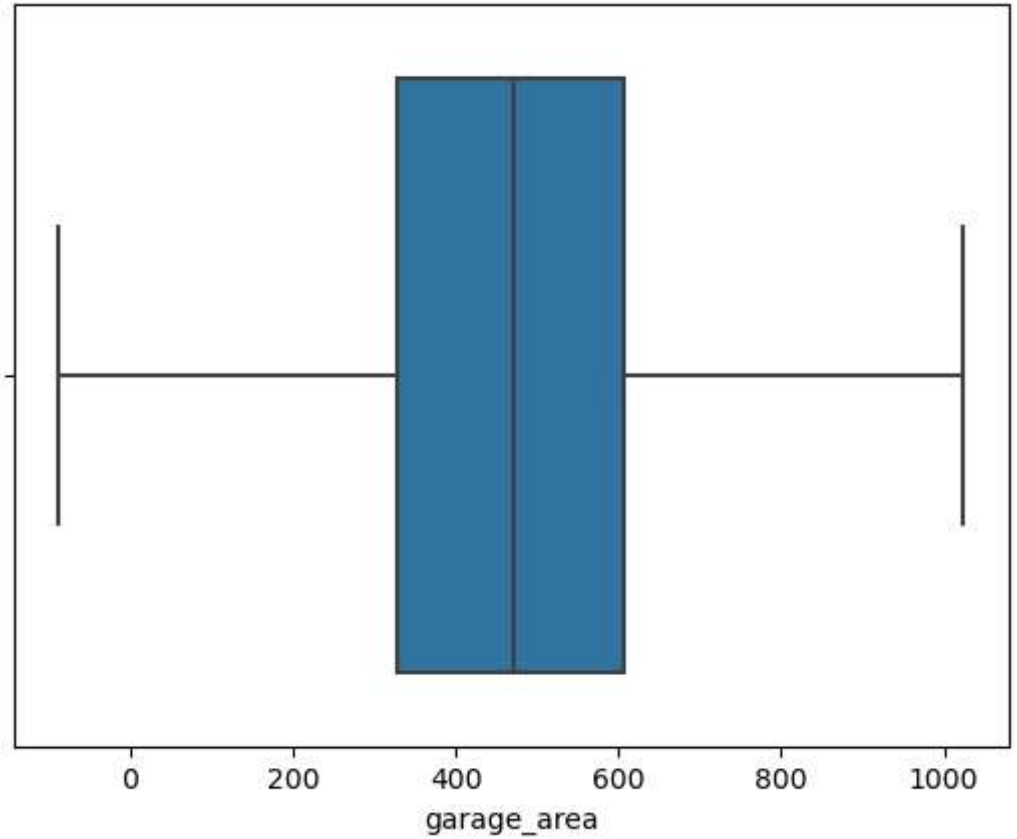
```
for col in ms_col:
    if col!='sale_price':
        q1=np.percentile(df[col],25)
        q3=np.percentile(df[col],75)
        iqr=q3-q1
        lowerfense=q1-(1.5*iqr)
        higherfense=q3+(1.5*iqr)
        df[col]=np.where(df[col]<lowerfense,lowerfense,df[col])
        df[col]=np.where(df[col]>higherfense,higherfense,df[col])
```

In [122…

```
## verify if any outliers in dataset
sns.boxplot(x= df['garage_area'])

## no any outliers found.
```

Out[122]:

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

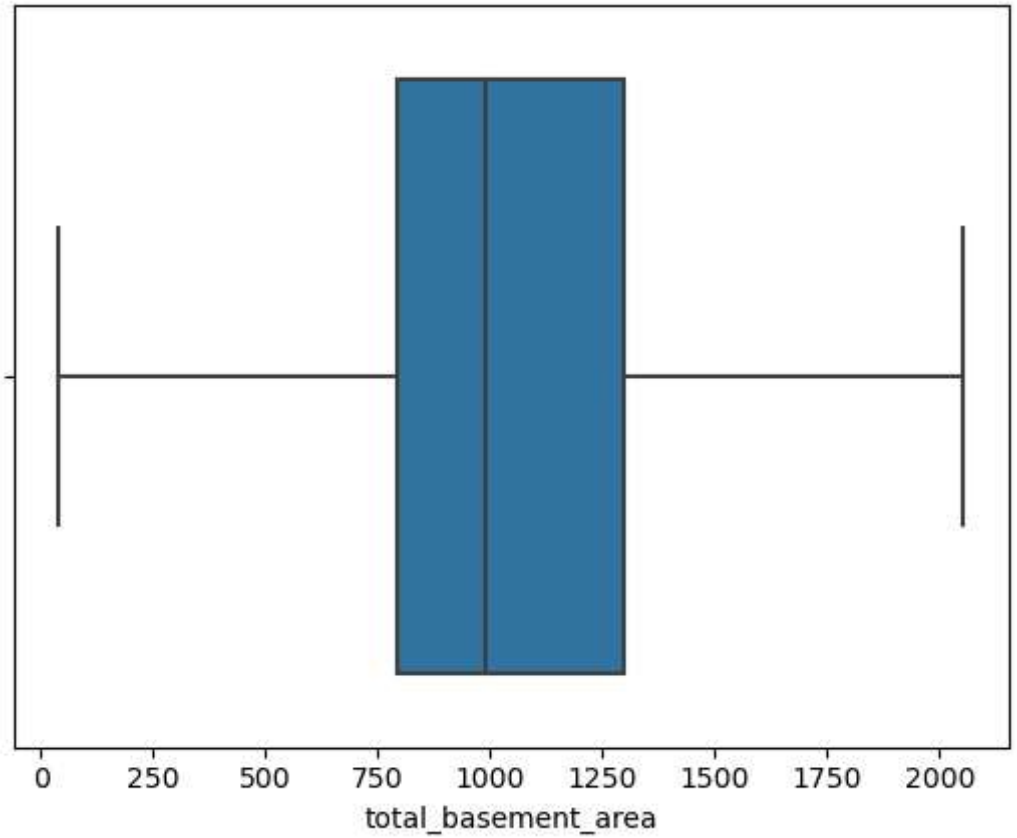


In [123…

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

## no any outliers found
```

Out[123]:

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



In [124…

```
## correlation in dataset
corrdf=df.corr()
corrdf
```

```
C:\Users\pv11379\AppData\Local\Temp\ipykernel_9800\3390713961.py:2: FutureWarning: The default value of numeric_only in
DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns or specify the v
alue of numeric_only to silence this warning.
  corrdf=df.corr()
```

Out[124]:

	house_condition	construction_year	remodel_year	bsmtfinsf1	total_basement_area	first_floor_area	second_floor_a
house_condition	1.000000	-0.399653	0.038757	-0.045718	-0.191959	-0.156679	0.007
construction_year	-0.399653	1.000000	0.594969	0.253017	0.410521	0.288461	0.011
remodel_year	0.038757	0.594969	1.000000	0.127007	0.302038	0.246918	0.139
bsmtfinsf1	-0.045718	0.253017	0.127007	1.000000	0.467215	0.395347	-0.157
total_basement_area	-0.191959	0.410521	0.302038	0.467215	1.000000	0.807059	-0.205
first_floor_area	-0.156679	0.288461	0.246918	0.395347	0.807059	1.000000	-0.226
second_floor_area	0.007627	0.011376	0.139562	-0.157207	-0.205889	-0.226931	1.000
lowqualfinsf	NaN	NaN	NaN	NaN	NaN	NaN	N
underground_full_bathroom	-0.052264	0.187782	0.120547	0.662899	0.303995	0.237080	-0.169
full_bathroom_above_grade	-0.225534	0.469853	0.438667	0.053543	0.332059	0.383804	0.420
bedroom_above_grade	0.005243	-0.055084	-0.034897	-0.111380	0.048827	0.128001	0.514
rooms_above_grade	-0.081303	0.106316	0.196536	0.017557	0.271134	0.401364	0.610
fireplaces	-0.037068	0.148045	0.111320	0.246820	0.330437	0.406630	0.194
garage_built_year	-0.334266	0.779114	0.616595	0.147448	0.320615	0.228015	0.068
garage_area	-0.017948	0.022637	0.009010	0.018886	0.019304	-0.008803	-0.038
pool_area	NaN	NaN	NaN	NaN	NaN	NaN	N
miscellaneous_value	NaN	NaN	NaN	NaN	NaN	NaN	N
year_sold	0.046120	-0.013273	0.035846	0.013887	-0.018578	-0.013570	-0.028
sale_price	-0.106139	0.524151	0.507015	0.400715	0.637244	0.620889	0.316

In [125…

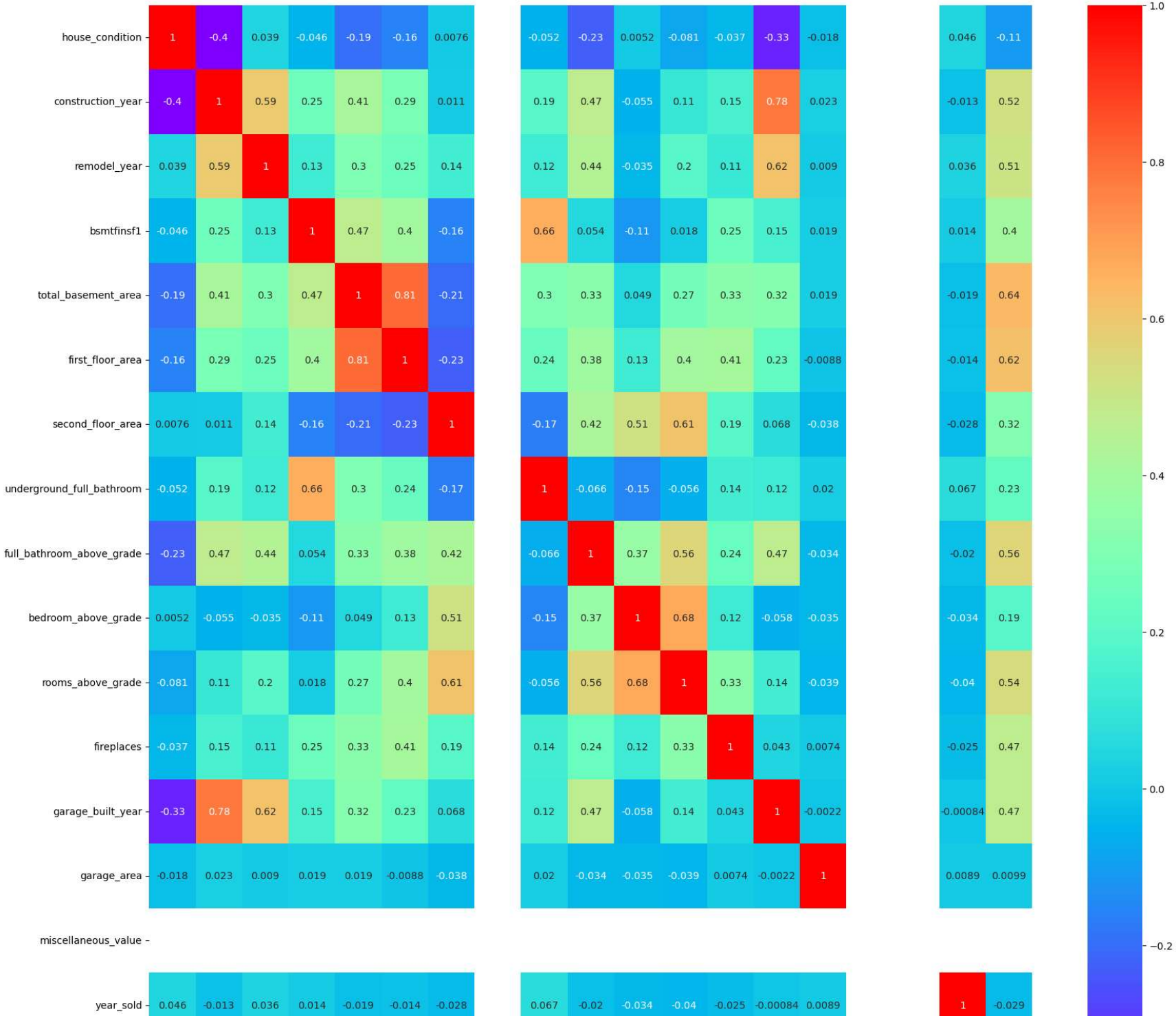
```
## drop Nan corelation column
corrdf.drop(['pool_area','lowqualfinsf'],inplace=True)
```

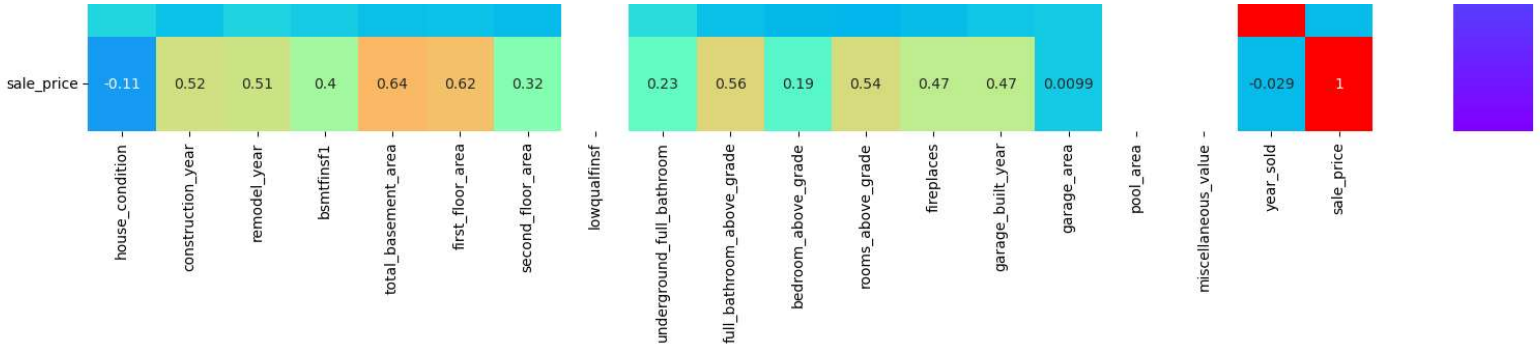
In [126…

```
plt.figure(figsize=(20,20))
sns.heatmap(corrdf,annot=True,cmap='rainbow')
## sale prize is having strong positive correlation with total basement area, construction and remodel year,first floor
```

Out[126]:

```
<Axes: >
```


house_condition 1 -0.4 0.039 -0.046 -0.19 -0.16 0.0076 -0.052 -0.23 0.0052 -0.081 -0.037 -0.33 -0.018 0.046 -0.11
construction_year -0.4 1 0.59 0.25 0.41 0.29 0.011 0.19 0.47 -0.055 0.11 0.15 0.78 0.023 -0.013 0.52
remodel_year 0.039 0.59 1 0.13 0.3 0.25 0.14 0.12 0.44 -0.035 0.2 0.11 0.62 0.009 0.036 0.51
bsmtfinsf1 -0.046 0.25 0.13 1 0.47 0.4 -0.16 0.66 0.054 -0.11 0.018 0.25 0.15 0.019 0.014 0.4
total_basement_area -0.19 0.41 0.3 0.47 1 0.81 -0.21 0.3 0.33 0.049 0.27 0.33 0.32 0.019 -0.019 0.64
first_floor_area -0.16 0.29 0.25 0.4 0.81 1 -0.23 0.24 0.38 0.13 0.4 0.41 0.23 -0.0088 -0.014 0.62
second_floor_area 0.0076 0.011 0.14 -0.16 -0.21 -0.23 1 -0.17 0.42 0.51 0.61 0.19 0.068 -0.038 -0.028 0.32
underground_full_bathroom -0.052 0.19 0.12 0.66 0.3 0.24 -0.17 1 -0.066 -0.15 -0.056 0.14 0.12 0.02 0.067 0.23
full_bathroom_above_grade -0.23 0.47 0.44 0.054 0.33 0.38 0.42 -0.066 1 0.37 0.56 0.24 0.47 -0.034 -0.02 0.56
bedroom_above_grade 0.0052 -0.055 -0.035 -0.11 0.049 0.13 0.51 -0.15 0.37 1 0.68 0.12 -0.058 -0.035 -0.034 0.19
rooms_above_grade -0.081 0.11 0.2 0.018 0.27 0.4 0.61 -0.056 0.56 0.68 1 0.33 0.14 -0.039 -0.04 0.54
fireplaces -0.037 0.15 0.11 0.25 0.33 0.41 0.19 0.14 0.24 0.12 0.33 1 0.043 0.0074 -0.025 0.47
garage_built_year -0.33 0.78 0.62 0.15 0.32 0.23 0.068 0.12 0.47 -0.058 0.14 0.043 1 -0.0022 -0.00084 0.47
garage_area -0.018 0.023 0.009 0.019 0.019 -0.0088 -0.038 0.02 -0.034 -0.035 -0.039 0.0074 -0.0022 1 0.0089 0.0099
miscellaneous_value
year_sold 0.046 -0.013 0.036 0.014 -0.019 -0.014 -0.028 0.067 -0.02 -0.034 -0.04 -0.025 -0.00084 0.0089 1 -0.029
1.0
0.8
0.6
0.4
0.2
0.0
−0.2



Road type property - Highest sale by volume

In [127…

```
df['road_type'].value_counts()
## paved road type is having high sale volume than gravel.
```

Out[127]:

```
Paved     1453
Gravel       6
Name: road_type, dtype: int64
```

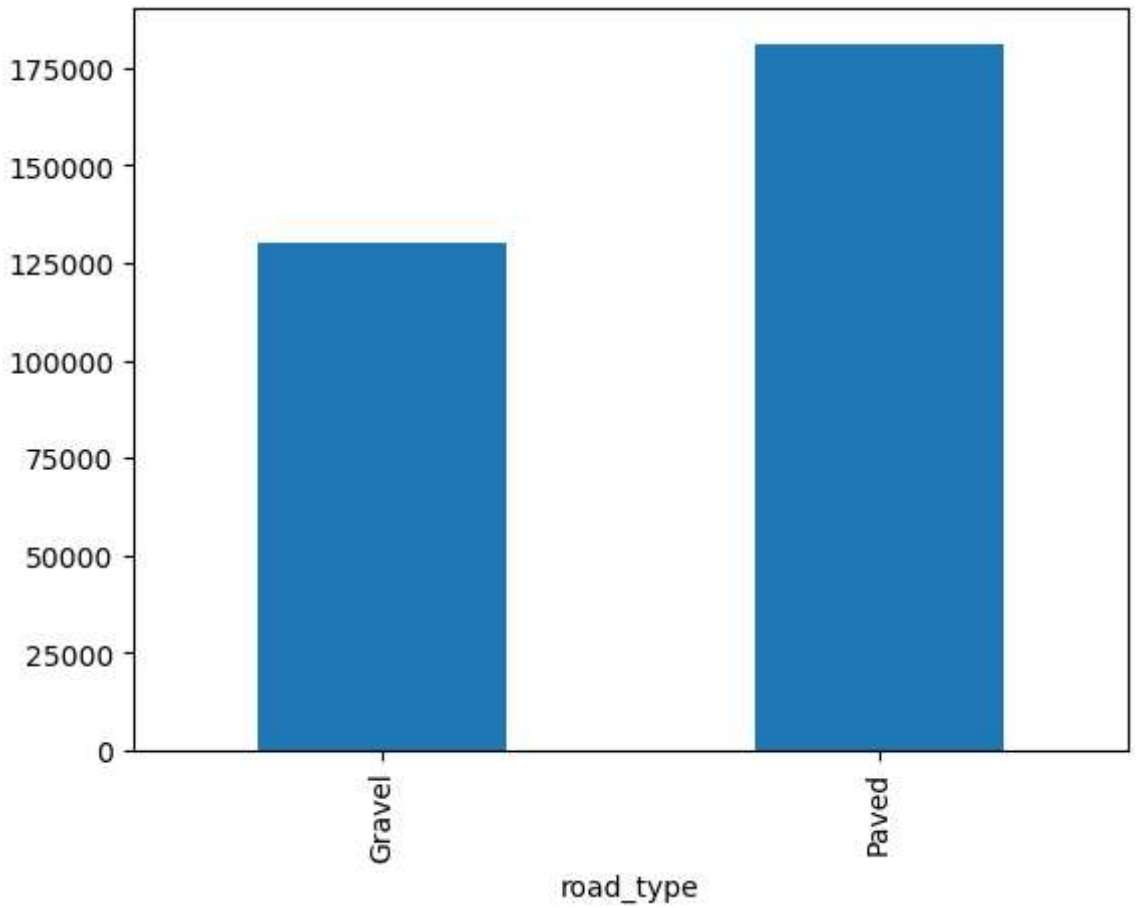
Road type property- Highest average sale price

In [128…

```
round(df.groupby('road_type')['sale_price'].mean(),0).plot(kind='bar')
## paved road type is having highest average sale price than gravel
```

Out[128]:

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



Property shape - highest sale by volume

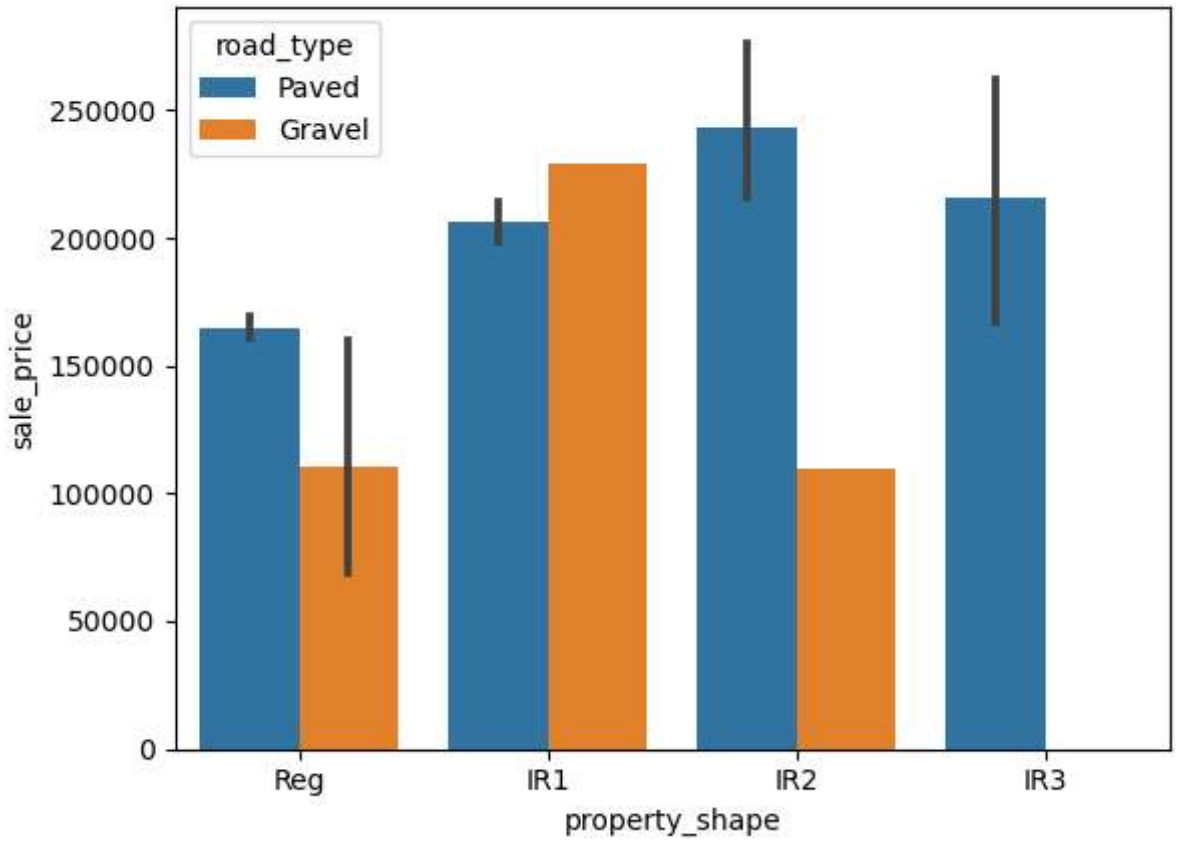
In [129…
```
df['property_shape'].value_counts()
## Reg property shape having highest sale volume
```

Out[129]:
```
Reg    924
IR1    484
IR2     41
IR3     10
Name: property_shape, dtype: int64
```

Property shape on which road type - Highest avg sale price

In [130…
```
sns.barplot(x=df['property_shape'],y=df['sale_price'],hue=df['road_type'])

## IR2 shape property on paved road having highest average sale price and IR2 on Gravel road lowest average sale price
```

Out[130]:
```
<Axes: xlabel='property_shape', ylabel='sale_price'>
```



Garage - Highest sale by volume

In [131…
```
df['garage'].value_counts()
## garage attached house having high selling volume.
```

Out[131]:
```
Attchd     950
Detchd     387
BuiltIn     88
Basment     19
CarPort      9
2TFes        5
2Types       1
Name: garage, dtype: int64
```

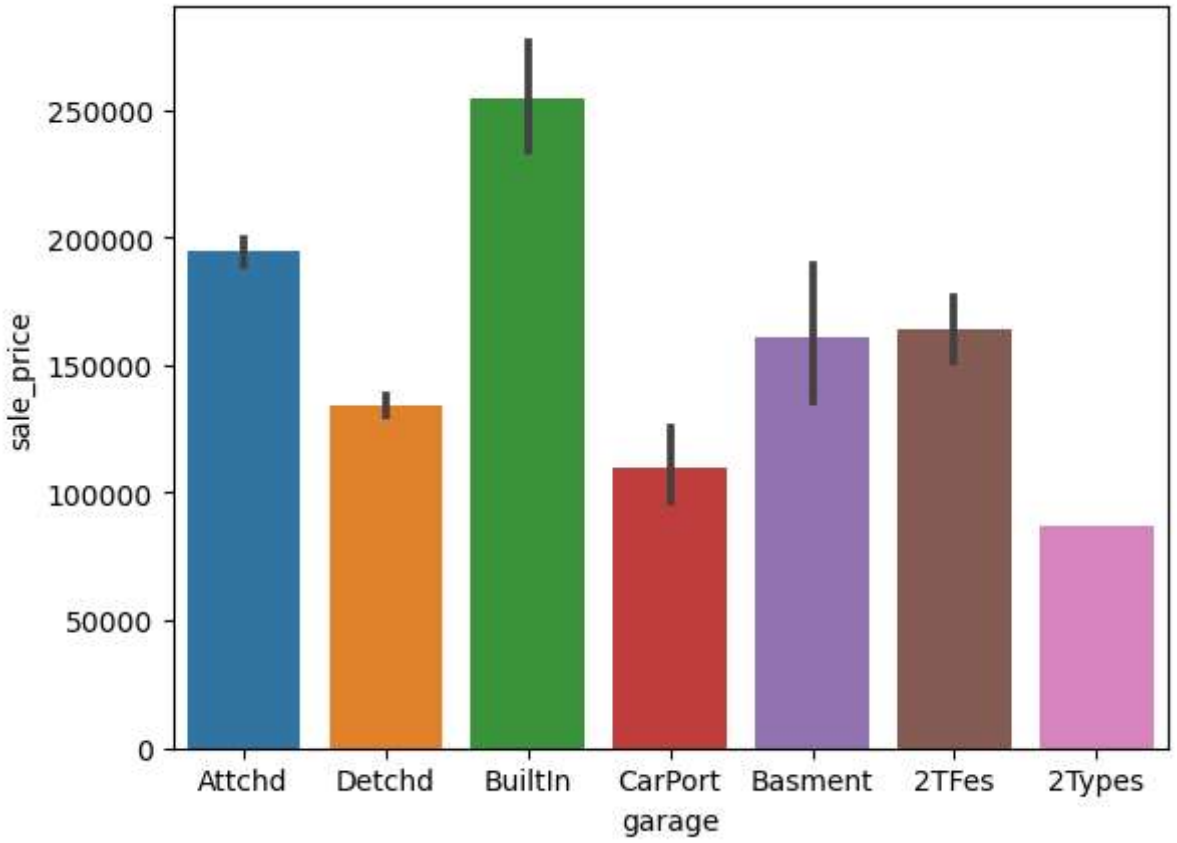
Garage - Highest avg sale price

In [132…
```
## average sale price wrt to garage
grg_saleprice=round(df.groupby('garage')['sale_price'].mean(),0)
pd.Series(grg_saleprice).sort_values(ascending=False)

## BuiltIn and Attchd garage is having hightest average sale price.
## BuiltIn - High sale price in low volume. So it is costly ever.
```

Out[132]:
```
garage
BuiltIn    254752.0
Attchd     194461.0
2TFes      164140.0
Basment    160571.0
Detchd     134091.0
CarPort    109962.0
2Types      87000.0
Name: sale_price, dtype: float64
```

In [133…
```
sns.barplot(x=df['garage'],y=df['sale_price'],estimator='mean')
```

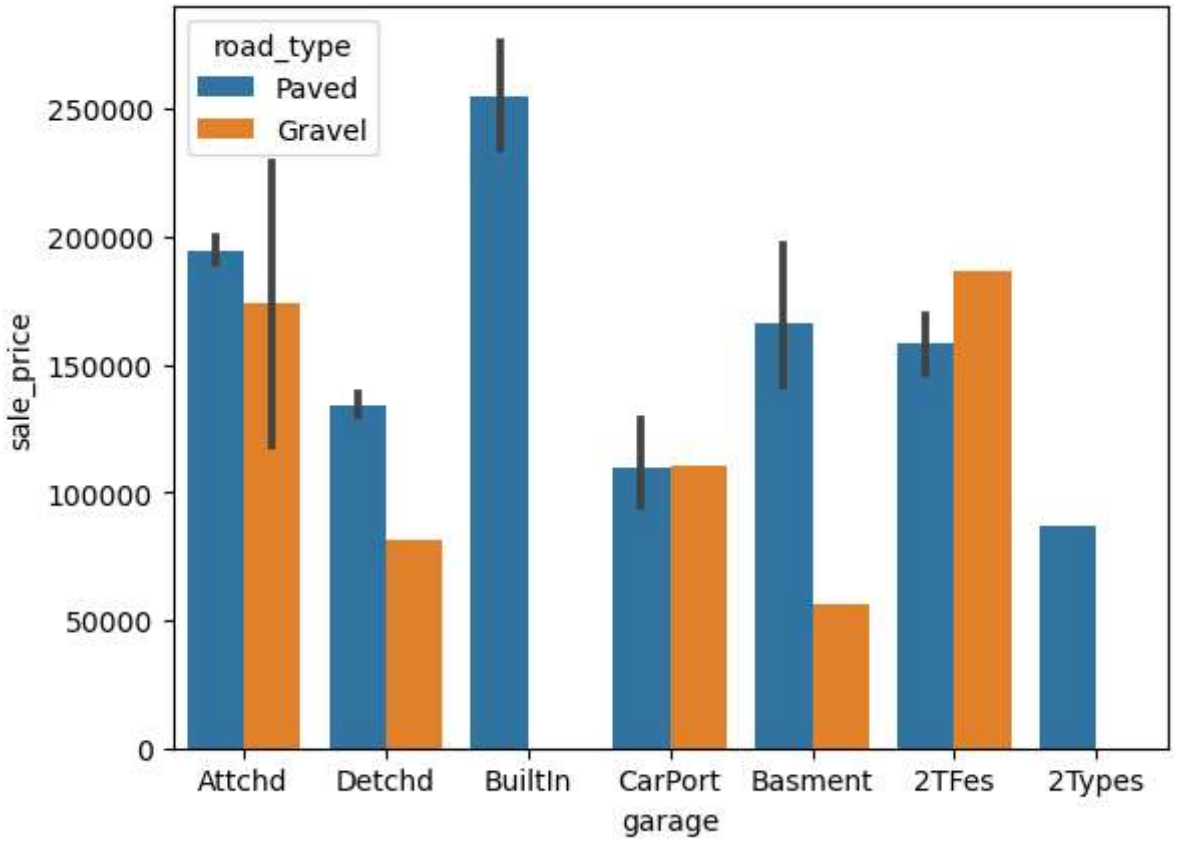
Out[133]:
```
<Axes: xlabel='garage', ylabel='sale_price'>
```



Garage on which road type - Highest avg sale price

In [134…
```
sns.barplot(x=df['garage'],hue=df['road_type'],y=df['sale_price'],estimator='mean')

## BuiltIn garage on paved road having highest aevrage sale price and basement on gravel road lowest average sale price
```

Out[134]:
```
<Axes: xlabel='garage', ylabel='sale_price'>
```



Garage on which road type and property shape - Highest avg sale price

In [139…

```
grp_saleprice=round(df.groupby(['garage','road_type','property_shape'])['sale_price'].mean().reset_index(),0)
grp_saleprice_highest=pd.DataFrame(grp_saleprice).sort_values(by=['sale_price'],ascending=False).head(5)
grp_saleprice_highest


### Builtin garage on paved road with IR2 shape having highest avg sale price.
```

Out[139]:

	garage	road_type	property_shape	sale_price
13	BuiltIn	Paved	IR2	296463.0
12	BuiltIn	Paved	IR1	261035.0
15	BuiltIn	Paved	Reg	244722.0
6	Attchd	Paved	IR2	240006.0
7	Attchd	Paved	IR3	234500.0

Garage on which road type and property shape - Lowest avg sale price

In [141…

```
grp_saleprice_lowest=pd.DataFrame(grp_saleprice).sort_values(by=['sale_price'],ascending=False).tail(5)
grp_saleprice_lowest

## Basement garage on gravel road with Reg shape having lowest avg sale price.
```

Out[141]:

	garage	road_type	property_shape	sale_price
18	CarPort	Paved	Reg	109276.0
2	2Types	Paved	Reg	87000.0
19	Detchd	Gravel	Reg	81000.0
22	Detchd	Paved	IR3	73000.0
9	Basment	Gravel	Reg	55993.0

House type - High sale volume

In [142…

```
df['house_type'].value_counts()
## 1Fam having high sale by volume.
```

Out[142]:

```
1Fam      1219
TwnhsE     114
Duplex      52
Twnhs       43
2fmCon      31
Name: house_type, dtype: int64
```

house_type - Highest avg sale price

In [144…
```
hrgptype_saleprice=round(df.groupby(['house_type','road_type','garage','property_shape'])['sale_price'].mean().reset_in
pd.DataFrame.sort_values(hrgptype_saleprice,by=['sale_price'],ascending=False).head(5)

## 1Fam on paved road with builtin garage and IR2 shape having highest sale price.
```

Out[144]:

	house_type	road_type	garage	property_shape	sale_price
12	1Fam	Paved	BuiltIn	IR2	296463.0
14	1Fam	Paved	BuiltIn	Reg	264929.0
11	1Fam	Paved	BuiltIn	IR1	262904.0
6	1Fam	Paved	Attchd	IR2	241546.0
7	1Fam	Paved	Attchd	IR3	234500.0

house_type - Lowest avg sale price

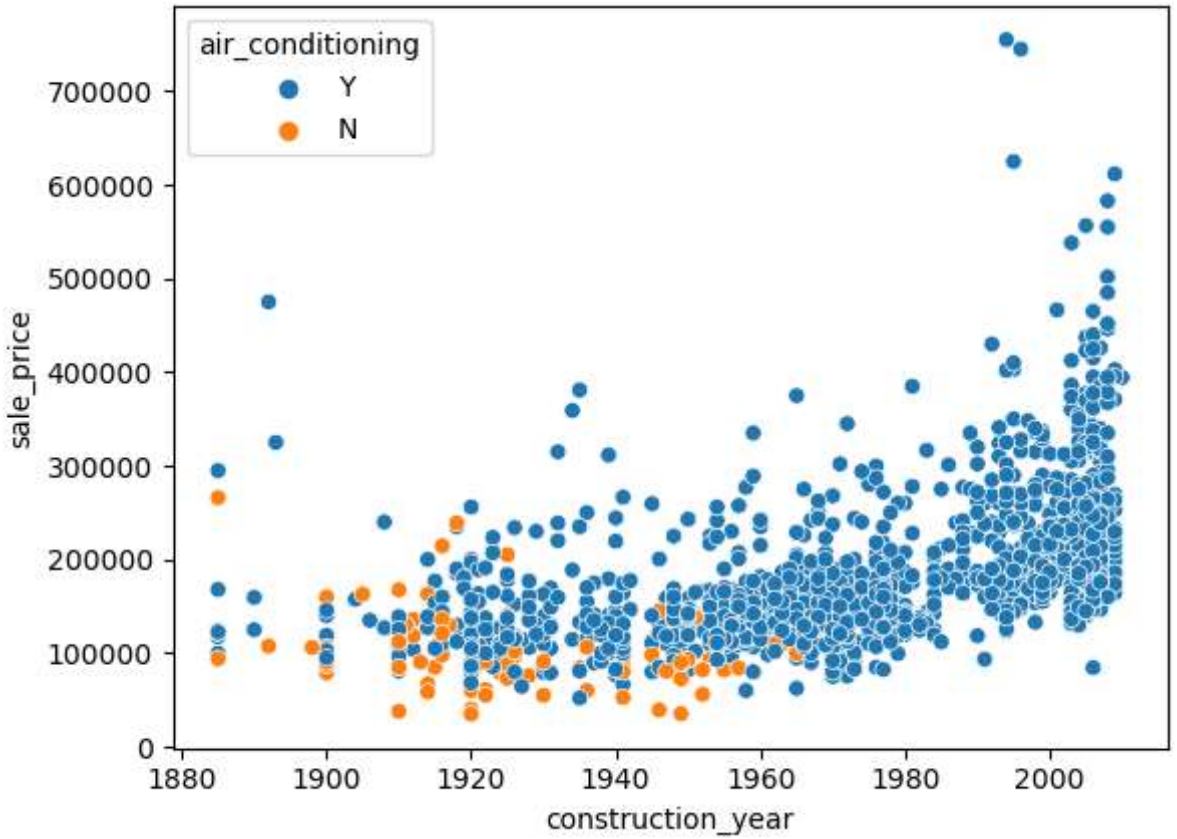
In [145…
```
pd.DataFrame.sort_values(hrgptype_saleprice,by=['sale_price'],ascending=False).tail(5)

## 1Fam on gravel road with basement garage and reg shape property having lowest sale price.
```

Out[145]:

	house_type	road_type	garage	property_shape	sale_price
24	2fmCon	Paved	BuiltIn	Reg	93000.0
4	1Fam	Paved	2Types	Reg	87000.0
2	1Fam	Gravel	Detchd	Reg	81000.0
19	1Fam	Paved	Detchd	IR3	73000.0
1	1Fam	Gravel	Basment	Reg	55993.0

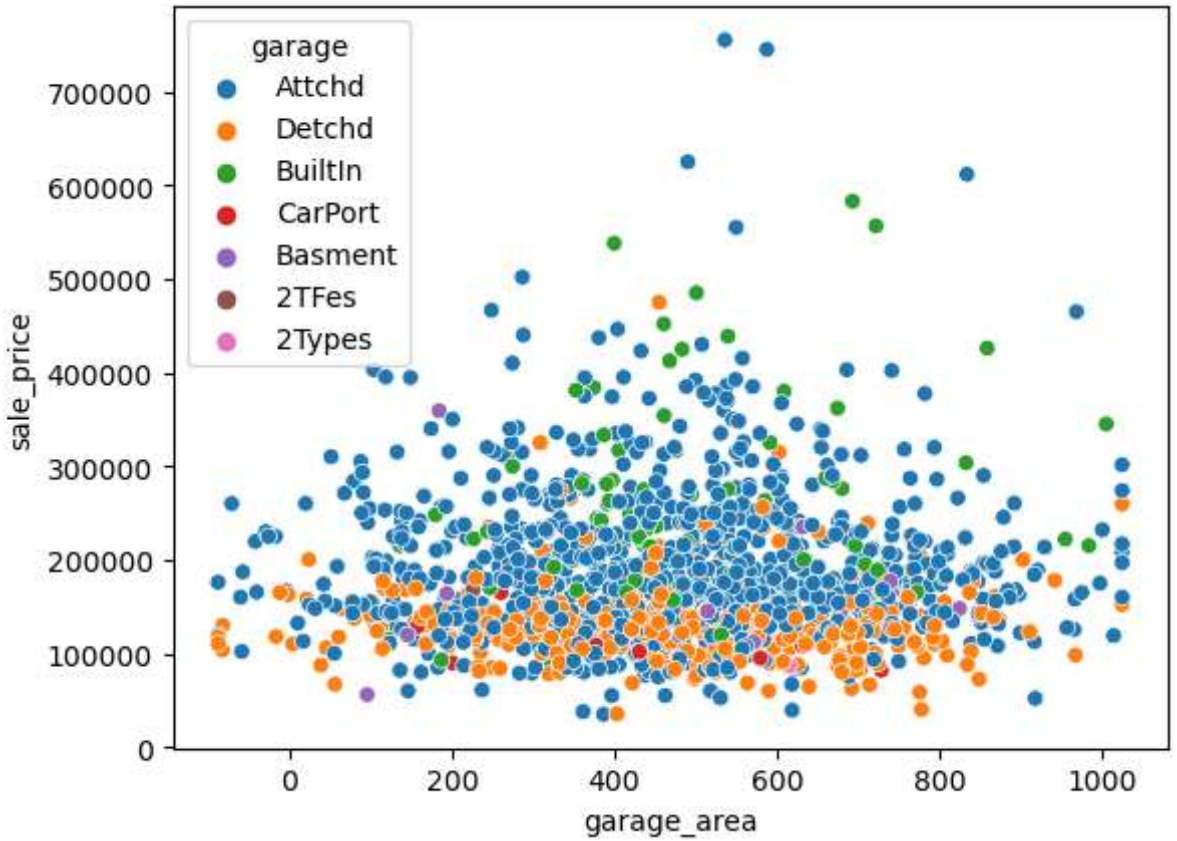
In [146…
```
sns.scatterplot(x=df['construction_year'],hue=df['air_conditioning'],y=df['sale_price'])
## sale price band is more on less aged air conditioned property.
```

Out[146]:
```
<Axes: xlabel='construction_year', ylabel='sale_price'>
```



In [147…
```
sns.scatterplot(x=df['garage_area'],hue=df['garage'],y=df['sale_price'])
```

Out[147]:
```
<Axes: xlabel='garage_area', ylabel='sale_price'>
```



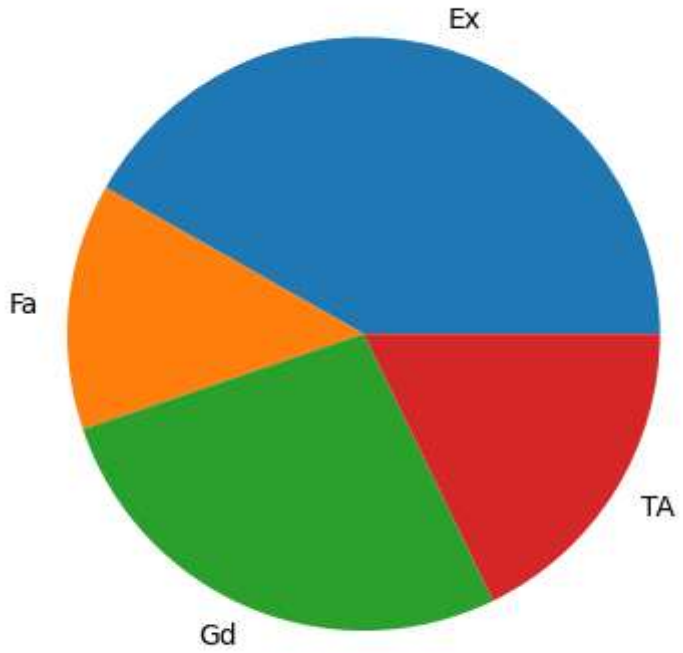
Sale price by kithcen quality

In [148…
```
kq_saleprice=round(df.groupby('kitchen_quality')['sale_price'].mean(),0)
kq_saleprice

## EX kitchen quality having highest avg sale price.
```

Out[148]:
```
kitchen_quality
Ex    328555.0
Fa    105565.0
Gd    212116.0
TA    139952.0
Name: sale_price, dtype: float64
```

In [149…
```
plt.pie(kq_saleprice.values,labels=kq_saleprice.index)
```

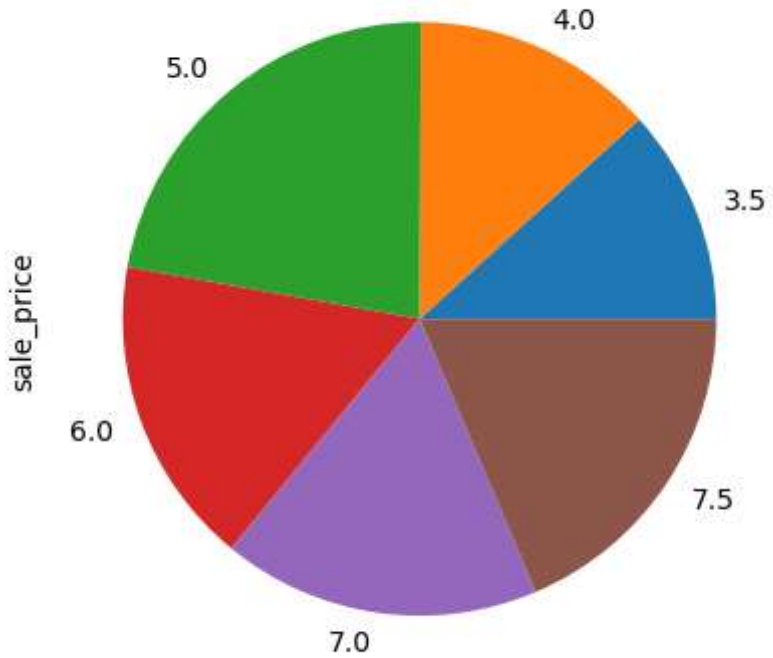
Out[149]:
```
([<matplotlib.patches.Wedge at 0x2ce2e1f1810>,
  <matplotlib.patches.Wedge at 0x2ce2e1f1f50>,
  <matplotlib.patches.Wedge at 0x2ce2e1f2ed0>,
  <matplotlib.patches.Wedge at 0x2ce2e1f3f50>],
 [Text(0.28055209444924323, 1.0636214186918873, 'Ex'),
  Text(-1.0951481554199676, 0.1032013453412443, 'Fa'),
  Text(-0.4236002642944289, -1.0151663982272512, 'Gd'),
  Text(0.9324210545515904, -0.583601728089371, 'TA')])
```



Sale price by House condition

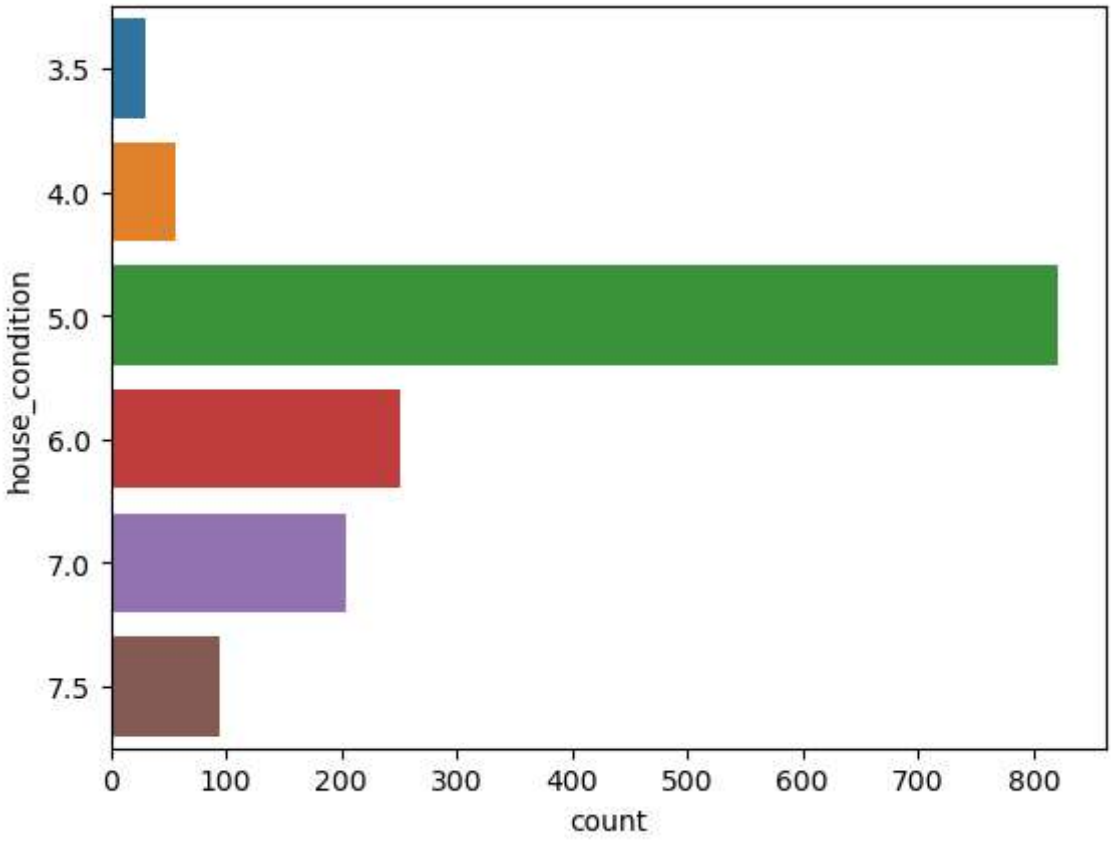
In [150…
```
hq_saleprice=round(df.groupby('house_condition')['sale_price'].mean(),0)
hq_saleprice.plot(kind='pie')

## 5 rating house condition - max avg sale price.
```

Out[150]:
```
<Axes: ylabel='sale_price'>
```



In [151…
```
sns.countplot(y=df['house_condition'])
## 5 rating house condition - max sale by volume
```

Out[151]:
```
<Axes: xlabel='count', ylabel='house_condition'>
```

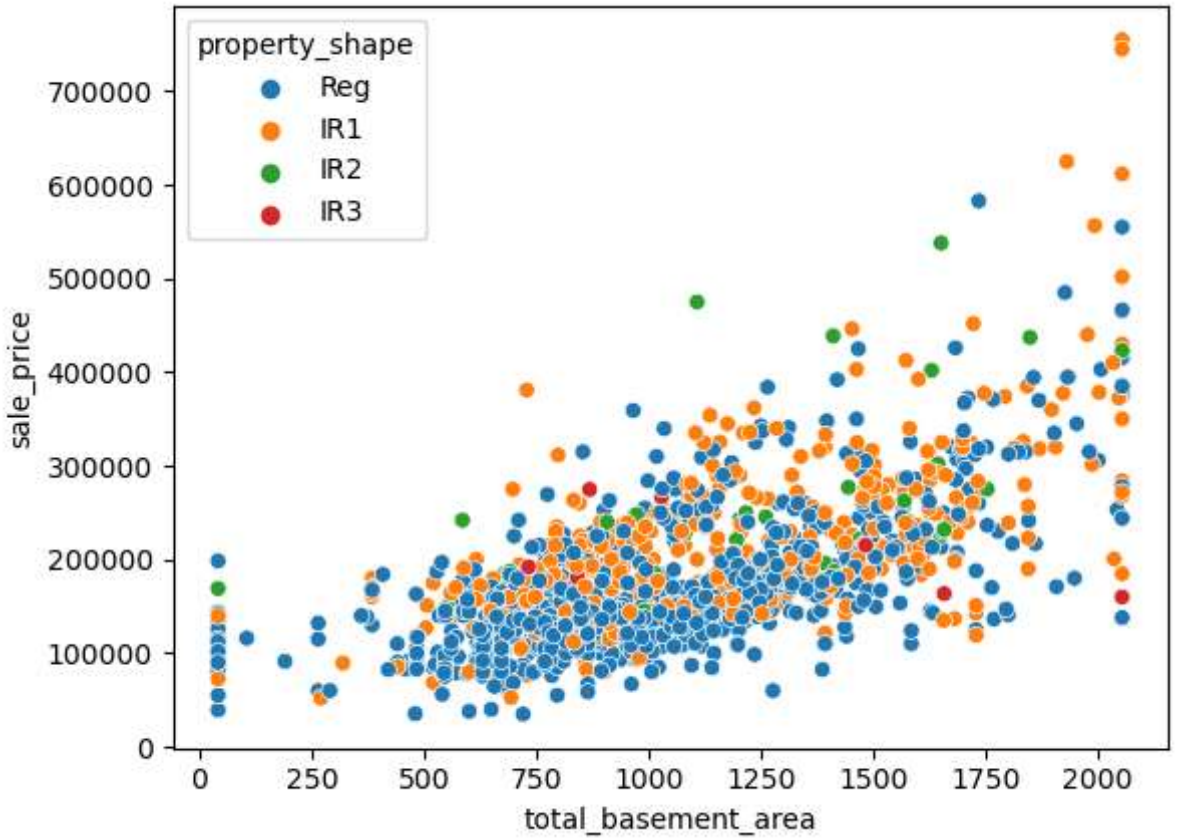


In [152…

```
sns.scatterplot(x=df['total_basement_area'],hue=df['property_shape'],y=df['sale_price'])

## Higher basemenet area-higher sale price
```

Out[152]:

```
<Axes: xlabel='total_basement_area', ylabel='sale_price'>
```



Property price analysis with Pool area

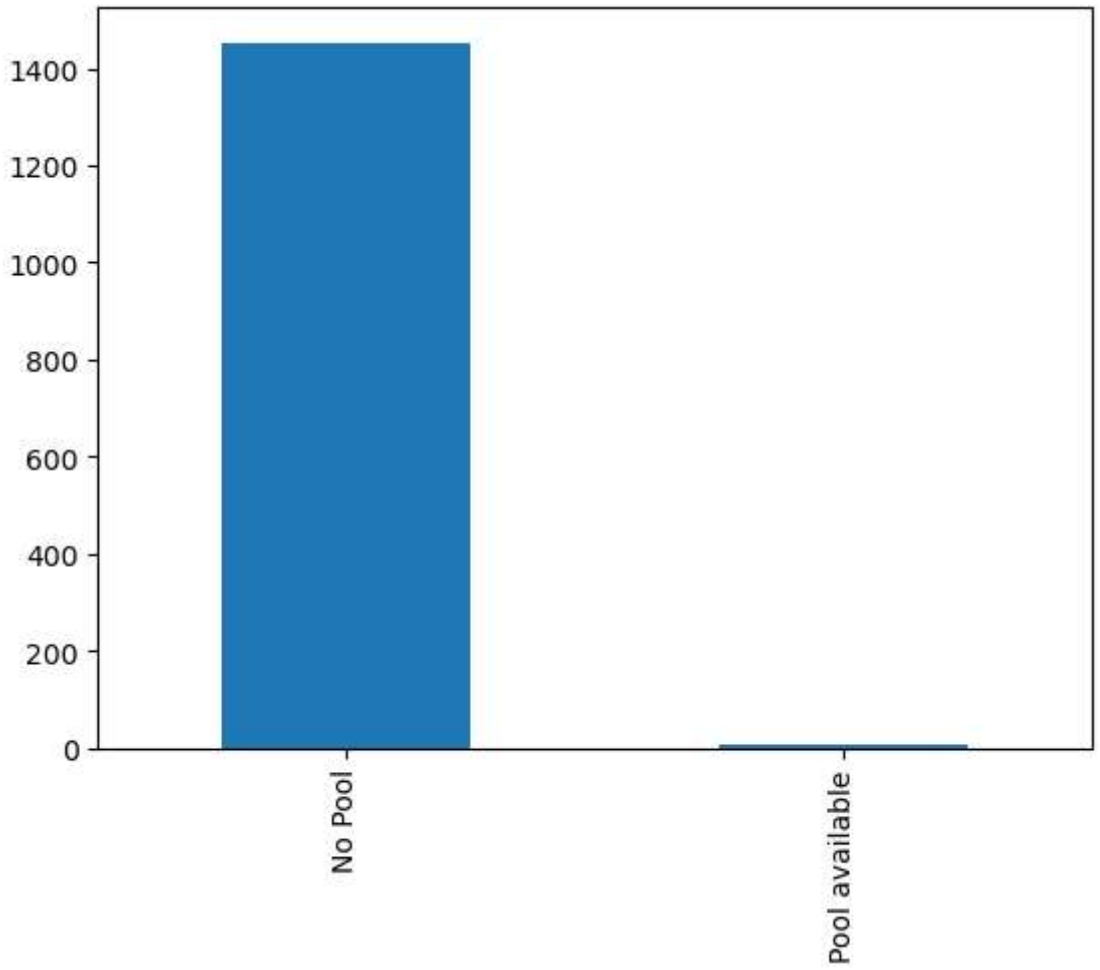
In [178…

```
## pool area analysis
data1=pd.read_csv('C:/Users/pv11379/Downloads/PropertyPrice_Data.csv')
data1['Pool_Area'].value_counts()

## No pool property having more volume.
```

Out[178]:
```
0      1452
512       1
648       1
576       1
555       1
480       1
519       1
738       1
Name: Pool_Area, dtype: int64
```

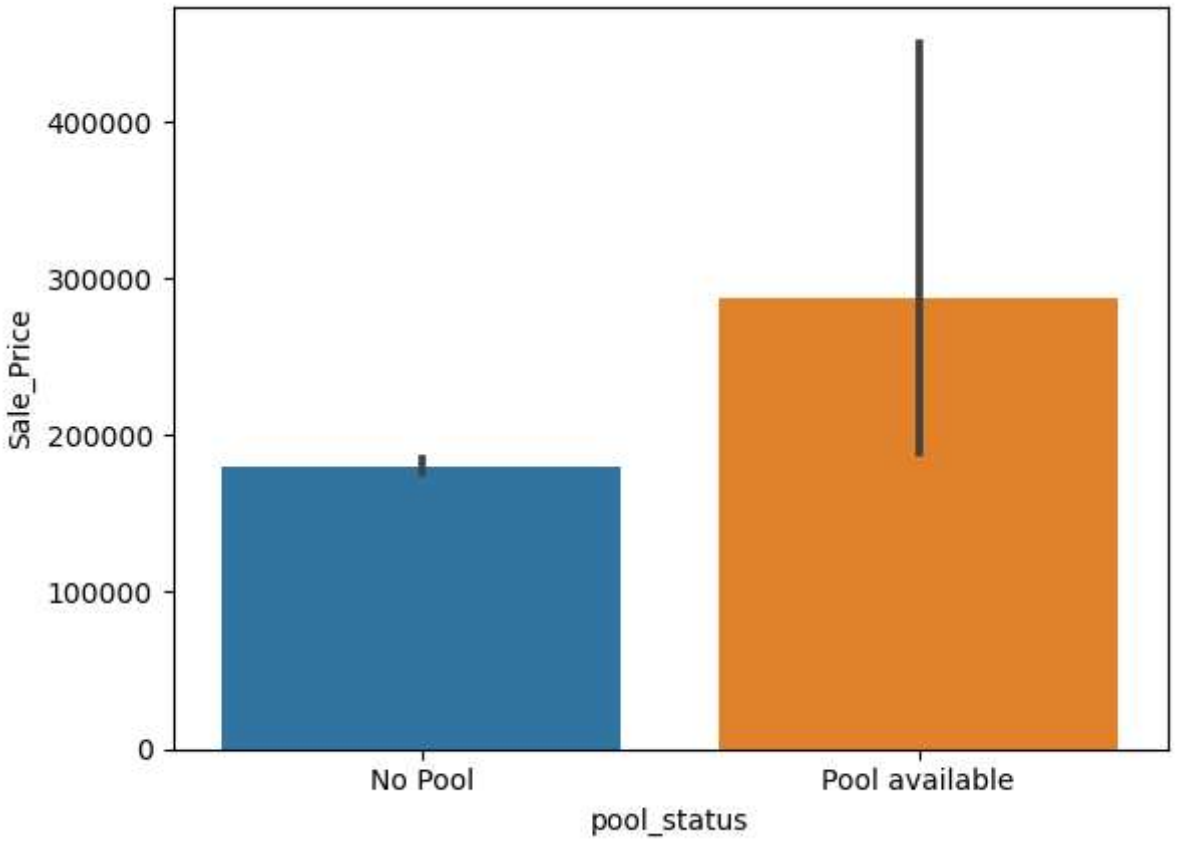
In [188…
```
data1['pool_status']=np.where(data1['Pool_Area']>0,'Pool available','No Pool')
data1['pool_status'].value_counts().plot(kind='bar')
```

Out[188]:
```
<Axes: >
```



In [193…
```
sns.barplot(x=data1['pool_status'],y=data1['Sale_Price'],estimator='mean')

## property with pool having max sale price than no pool property.
```

Out[193]:
```
<Axes: xlabel='pool_status', ylabel='Sale_Price'>
```



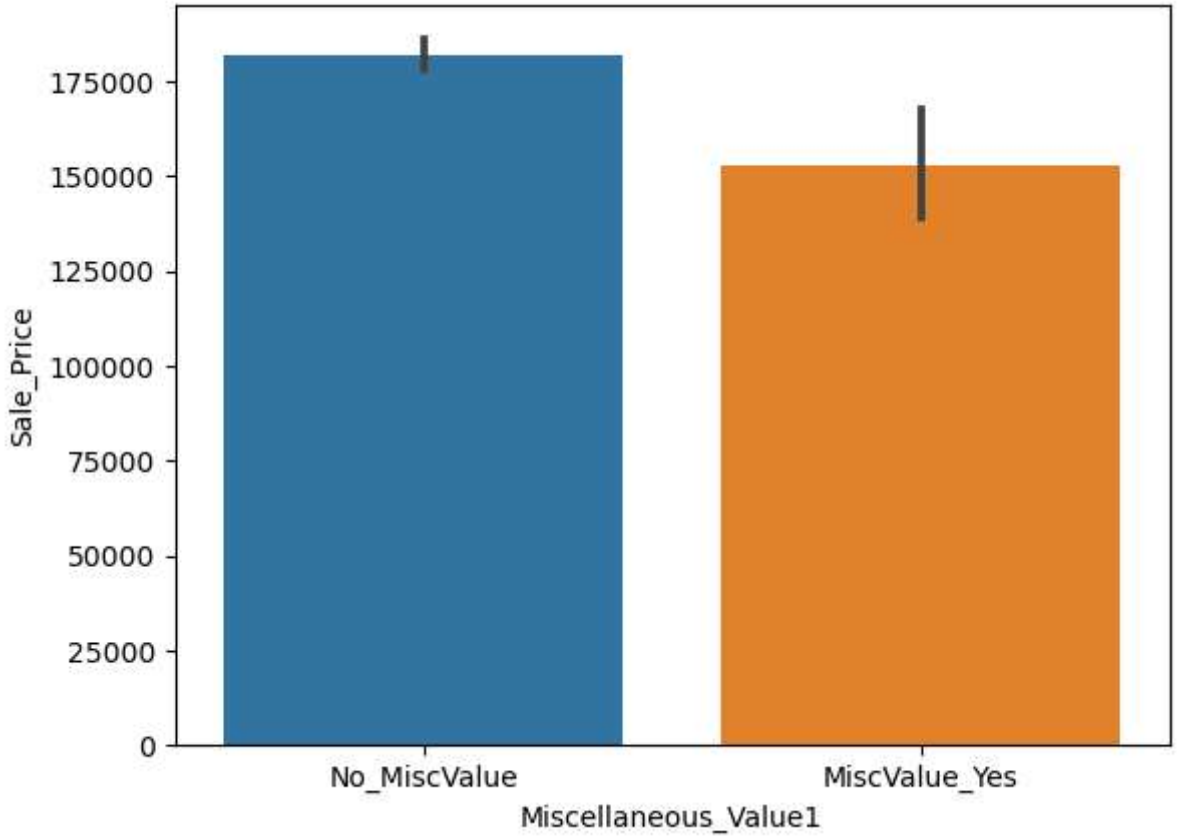
Sale price with Miscellaneous value

In [197…
```
data['Miscellaneous_Value1']=np.where(data['Miscellaneous_Value']>0,'MiscValue_Yes','No_MiscValue')
data['Miscellaneous_Value1'].value_counts()

## 1407 property having no miscellanous values.
```

Out[197]:
```
No_MiscValue     1407
MiscValue_Yes      52
Name: Miscellaneous_Value1, dtype: int64
```

In [201…
```
sns.barplot(x=data['Miscellaneous_Value1'],y=data['Sale_Price'],estimator='mean')

## No major difference in sale price of miscellaneous and no miscellanous property.
```

Out[201]:
```
<Axes: xlabel='Miscellaneous_Value1', ylabel='Sale_Price'>
```

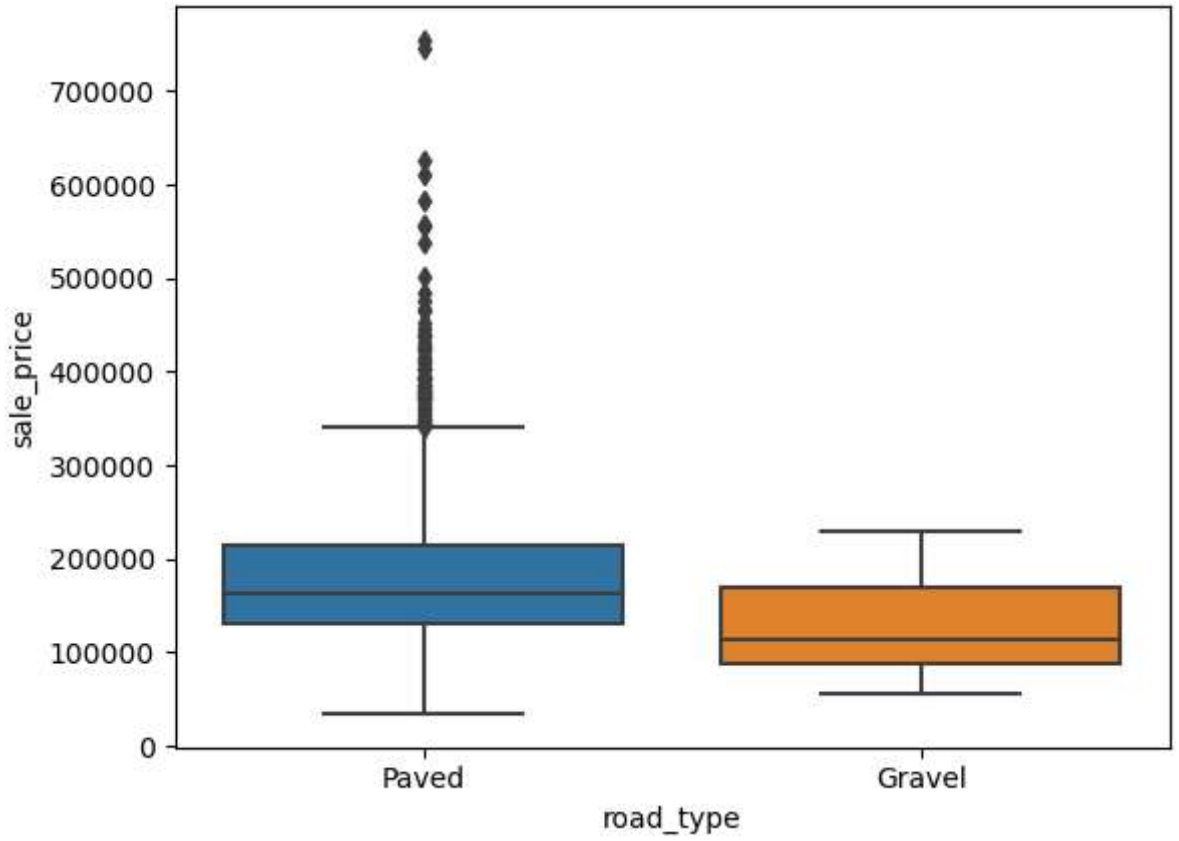


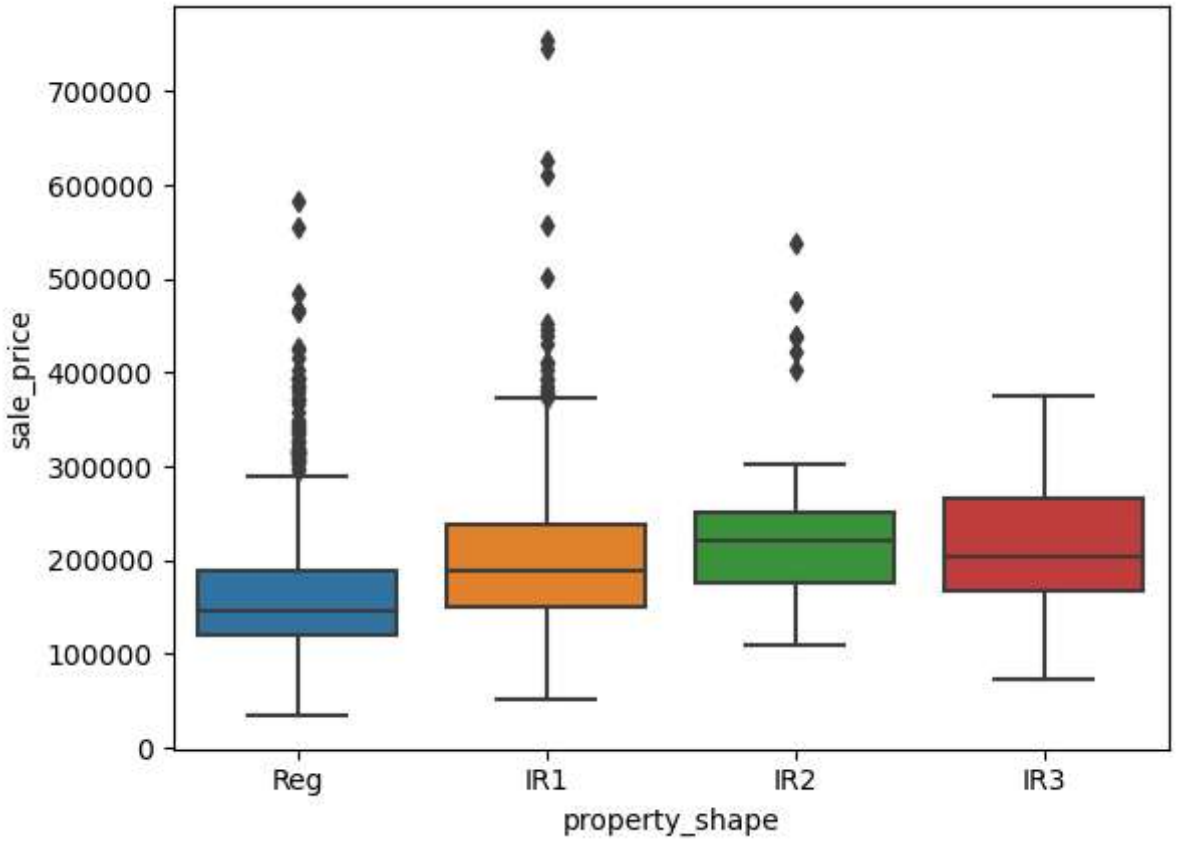
one point analysis of sale price with categorial parameter

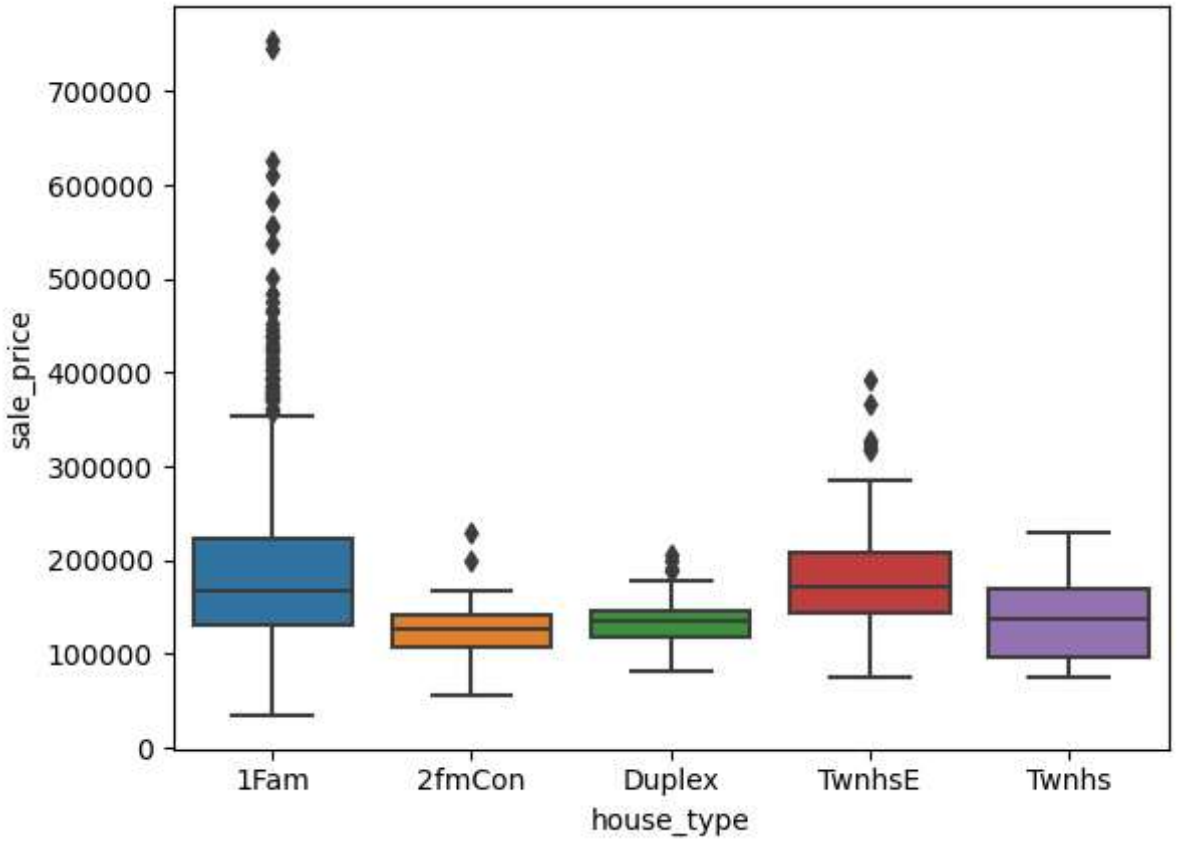
In [208…
```
## get all categorial parameter
cat_var=df.columns[df.dtypes=='O']
cat_var
```

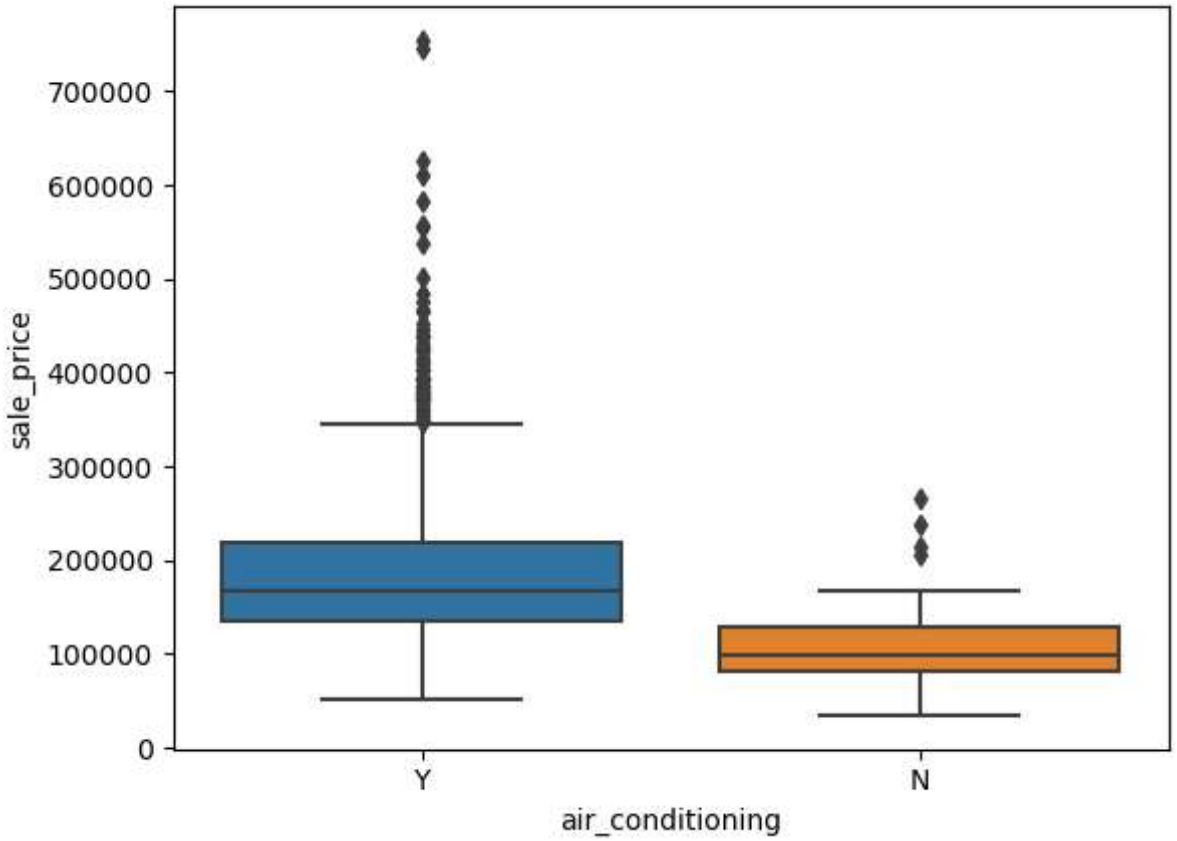
Out[208]:
```
Index(['road_type', 'property_shape', 'house_type', 'air_conditioning',
       'kitchen_quality', 'garage'],
      dtype='object')
```

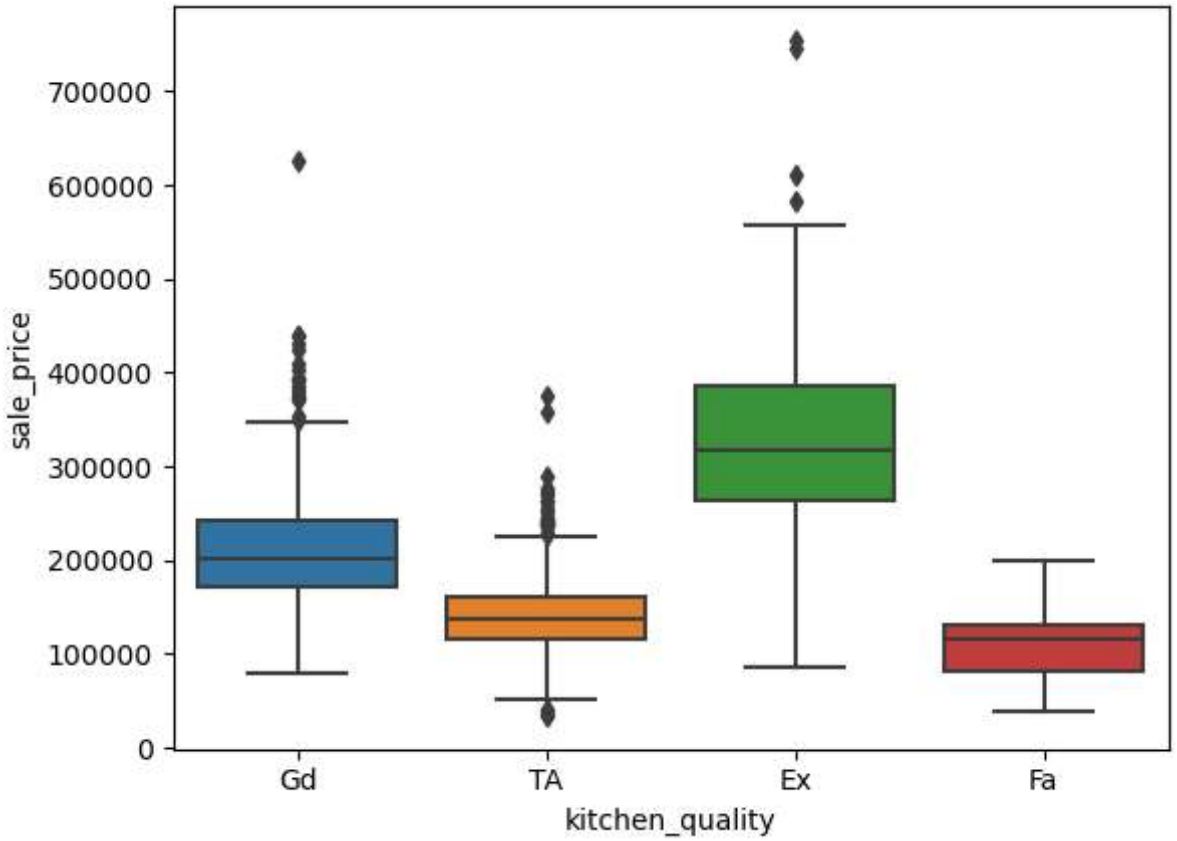
In [211…
```
for i in cat_var:
    plt.figure()
    sns.boxplot(x=df[i],y=df['sale_price'])

## easily get analysed in all below box plot which was anlaysed in earlier analysis.
```

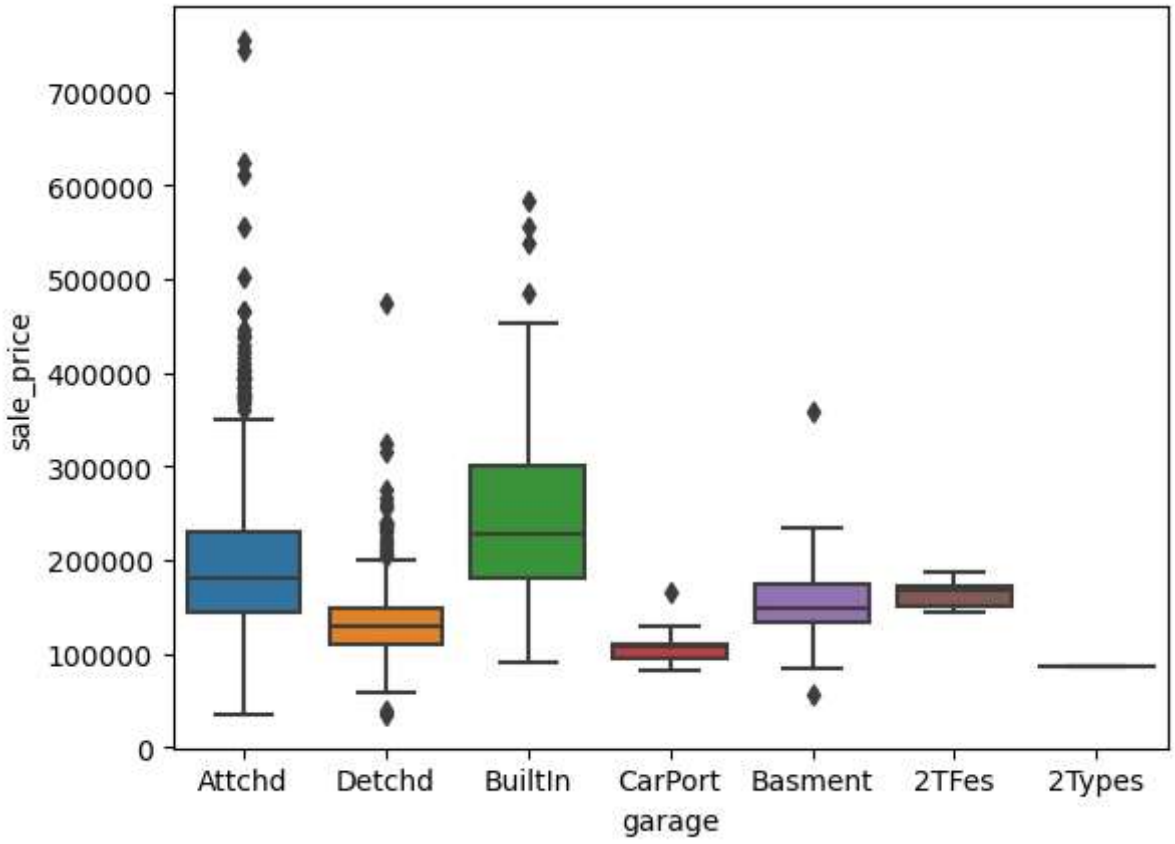



700000
600000
500000
400000
300000
200000
100000
0
sale_price
Reg
IR1
IR2
IR3
property_shape


700000
600000
500000
400000
300000
200000
100000
0
sale_price
1Fam
2fmCon
Duplex
TwnhsE
Twnhs
house_type


700000
600000
500000
400000
300000
200000
100000
0
sale_price
Y
N
air_conditioning


700000
600000
500000
400000
300000
200000
100000
0
sale_price
Gd
TA
Ex
Fa
kitchen_quality


700000
600000
500000
400000
300000
200000
100000
0
sale_price
Attchd
Detchd
BuiltIn
CarPort
Basment
2TFes
2Types
garage

In [ ]: