In [70]:
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score
from sklearn.metrics import accuracy_score
import warnings
warnings.filterwarnings('ignore')
```

In [2]:
```python
house_price =pd.read_csv("train.csv")
```

In [3]:
```python
house_price.head()
```

Out[3]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	YrSold	SaleType
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2008	WD
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	5	2007	WD
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	9	2008	WD
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2006	WD
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	12	2008	WD

5 rows × 81 columns

In [4]:
```python
# Data pre-processiong
house_price.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Id             1460 non-null   int64
 1   MSSubClass     1460 non-null   int64
 2   MSZoning       1460 non-null   object
 3   LotFrontage    1201 non-null   float64
 4   LotArea        1460 non-null   int64
 5   Street         1460 non-null   object
 6   Alley          91 non-null     object
 7   LotShape       1460 non-null   object
 8   LandContour    1460 non-null   object
 9   Utilities      1460 non-null   object
 10  LotConfig      1460 non-null   object
 11  LandSlope      1460 non-null   object
 12  Neighborhood   1460 non-null   object
 13  Condition1     1460 non-null   object
 14  Condition2     1460 non-null   object
 15  BldgType       1460 non-null   object
 16  HouseStyle     1460 non-null   object
 17  OverallQual    1460 non-null   int64
 18  OverallCond    1460 non-null   int64
 19  YearBuilt      1460 non-null   int64
 20  YearRemodAdd   1460 non-null   int64
 21  RoofStyle      1460 non-null   object
 22  RoofMatl       1460 non-null   object
 23  Exterior1st    1460 non-null   object
 24  Exterior2nd    1460 non-null   object
 25  MasVnrType     1452 non-null   object
 26  MasVnrArea     1452 non-null   float64
 27  ExterQual      1460 non-null   object
 28  ExterCond      1460 non-null   object
 29  Foundation     1460 non-null   object
 30  BsmtQual       1423 non-null   object
 31  BsmtCond       1423 non-null   object
 32  BsmtExposure   1422 non-null   object
 33  BsmtFinType1   1423 non-null   object
 34  BsmtFinSF1     1460 non-null   int64
 35  BsmtFinType2   1422 non-null   object
 36  BsmtFinSF2     1460 non-null   int64
 37  BsmtUnfSF      1460 non-null   int64
 38  TotalBsmtSF    1460 non-null   int64
 39  Heating        1460 non-null   object
 40  HeatingQC      1460 non-null   object
 41  CentralAir     1460 non-null   object
 42  Electrical     1459 non-null   object
 43  1stFlrSF       1460 non-null   int64
 44  2ndFlrSF       1460 non-null   int64
 45  LowQualFinSF   1460 non-null   int64
 46  GrLivArea      1460 non-null   int64
 47  BsmtFullBath   1460 non-null   int64
 48  BsmtHalfBath   1460 non-null   int64
 49  FullBath       1460 non-null   int64
 50  HalfBath       1460 non-null   int64
 51  BedroomAbvGr   1460 non-null   int64
 52  KitchenAbvGr   1460 non-null   int64
 53  KitchenQual    1460 non-null   object
 54  TotRmsAbvGrd   1460 non-null   int64
 55  Functional     1460 non-null   object
 56  Fireplaces     1460 non-null   int64
 57  FireplaceQu    770 non-null    object
 58  GarageType     1379 non-null   object
 59  GarageYrBlt    1379 non-null   float64
 60  GarageFinish   1379 non-null   object
 61  GarageCars     1460 non-null   int64
 62  GarageArea     1460 non-null   int64
 63  GarageQual     1379 non-null   object
 64  GarageCond     1379 non-null   object
 65  PavedDrive     1460 non-null   object
 66  WoodDeckSF     1460 non-null   int64
 67  OpenPorchSF    1460 non-null   int64
 68  EnclosedPorch  1460 non-null   int64
 69  3SsnPorch      1460 non-null   int64
 70  ScreenPorch    1460 non-null   int64
 71  PoolArea       1460 non-null   int64
 72  PoolQC         7 non-null      object
 73  Fence          281 non-null    object
 74  MiscFeature    54 non-null     object
 75  MiscVal        1460 non-null   int64
 76  MoSold         1460 non-null   int64
 77  YrSold         1460 non-null   int64
 78  SaleType       1460 non-null   object
 79  SaleCondition  1460 non-null   object
 80  SalePrice      1460 non-null   int64
dtypes: float64(3), int64(35), object(43)
memory usage: 924.0+ KB
```

In [5]:
```python
house_price.describe()
```

Out[5]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	...	WoodDeckSF	OpenPorchSF
count	1460.000000	1460.000000	1201.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1452.000000	1460.000000	...	1460.000000	1460.000000
mean	730.500000	56.897260	70.049958	10516.828082	6.099315	5.575342	1971.267808	1984.865753	103.685262	443.639726	...	94.244521	46.660274
std	421.610009	42.300571	24.284752	9981.264932	1.382997	1.112799	30.202904	20.645407	181.066207	456.098091	...	125.338794	66.256028
min	1.000000	20.000000	21.000000	1300.000000	1.000000	1.000000	1872.000000	1950.000000	0.000000	0.000000	...	0.000000	0.000000
25%	365.750000	20.000000	59.000000	7553.500000	5.000000	5.000000	1954.000000	1967.000000	0.000000	0.000000	...	0.000000	0.000000
50%	730.500000	50.000000	69.000000	9478.500000	6.000000	5.000000	1973.000000	1994.000000	0.000000	383.500000	...	0.000000	25.000000
75%	1095.250000	70.000000	80.000000	11601.500000	7.000000	6.000000	2000.000000	2004.000000	166.000000	712.250000	...	168.000000	68.000000
max	1460.000000	190.000000	313.000000	215245.000000	10.000000	9.000000	2010.000000	2010.000000	1600.000000	5644.000000	...	857.000000	547.000000

8 rows × 38 columns

In [6]:
```python
house_price.columns
```

Out[6]:
```
Index(['Id', 'MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Street',
       'Alley', 'LotShape', 'LandContour', 'Utilities', 'LotConfig',
       'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'BldgType',
       'HouseStyle', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd',
       'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType',
       'MasVnrArea', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual',
       'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1',
       'BsmtFinType2', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating',
       'HeatingQC', 'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF',
       'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath',
       'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual',
       'TotRmsAbvGrd', 'Functional', 'Fireplaces', 'FireplaceQu', 'GarageType',
       'GarageYrBlt', 'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQual',
       'GarageCond', 'PavedDrive', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'PoolQC',
       'Fence', 'MiscFeature', 'MiscVal', 'MoSold', 'YrSold', 'SaleType',
       'SaleCondition', 'SalePrice'],
      dtype='object')
```

In [7]:
```python
house_price.shape
```

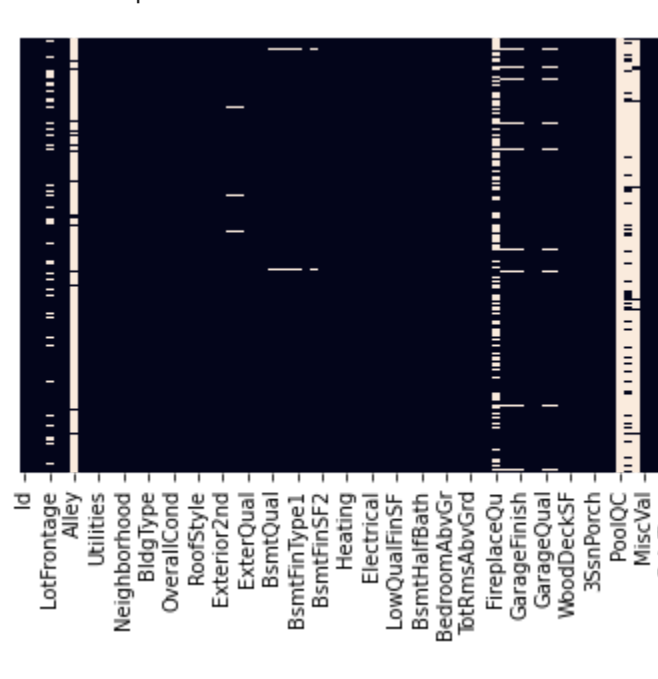
Out[7]: (1460, 81)

In [8]:
```python
#how many null vallues is columns
house_price.isnull().sum()
```

Out[8]:
```
Id                 0
MSSubClass         0
MSZoning           0
LotFrontage      259
LotArea            0
                ...
MoSold             0
YrSold             0
SaleType           0
SaleCondition      0
SalePrice          0
Length: 81, dtype: int64
```

In [9]:
```python
sns.heatmap(house_price.isnull(),yticklabels=False,cbar=False)
```

Out[9]: <AxesSubplot:>



In [10]:
```python
#fill all missing values
house_price['LotFrontage']=house_price['LotFrontage'].fillna(house_price['LotFrontage'].mean())
```

In [11]:
```python
house_price.drop(['Alley'],axis=1,inplace=True)
```

In [12]:
```python
house_price['BsmtCond']=house_price['BsmtCond'].fillna(house_price['BsmtCond'].mode()[0])
house_price['BsmtQual']=house_price['BsmtQual'].fillna(house_price['BsmtQual'].mode()[0])
```

In [13]:
```python
house_price['FireplaceQu']=house_price['FireplaceQu'].fillna(house_price['FireplaceQu'].mode()[0])
house_price['GarageType']=house_price['GarageType'].fillna(house_price['GarageType'].mode()[0])
```

In [14]:
```python
house_price.drop(['GarageYrBlt'],axis=1,inplace=True)
```

In [15]:
```python
house_price['GarageFinish']=house_price['GarageFinish'].fillna(house_price['GarageFinish'].mode()[0])
house_price['GarageQual']=house_price['GarageQual'].fillna(house_price['GarageQual'].mode()[0])
house_price['GarageCond']=house_price['GarageCond'].fillna(house_price['GarageCond'].mode()[0])
```

In [16]:
```python
house_price.drop(['PoolQC','Fence','MiscFeature'],axis=1,inplace=True)
```

In [17]:
```python
house_price.shape
```

Out[17]: (1460, 76)

In [18]:
```python
#Remove all null values
house_price.drop(['Id'],axis=1,inplace=True)
```

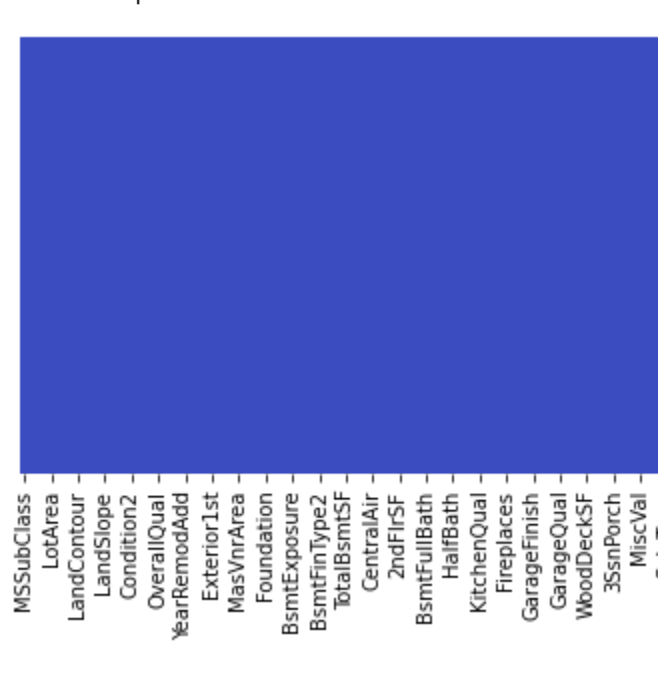
In [19]:
```python
house_price.isnull().sum()
```

Out[19]:
```
MSSubClass       0
MSZoning         0
LotFrontage      0
LotArea          0
Street           0
                ..
MoSold           0
YrSold           0
SaleType         0
SaleCondition    0
SalePrice        0
Length: 75, dtype: int64
```

In [71]:
```python
house_price['MasVnrType']=house_price['MasVnrType'].fillna(house_price['MasVnrType'].mode()[0])
house_price['MasVnrArea']=house_price['MasVnrArea'].fillna(house_price['MasVnrArea'].mode()[0])
```

In [72]:
```python
sns.heatmap(house_price.isnull(),yticklabels=False,cbar=False,cmap='coolwarm')
```

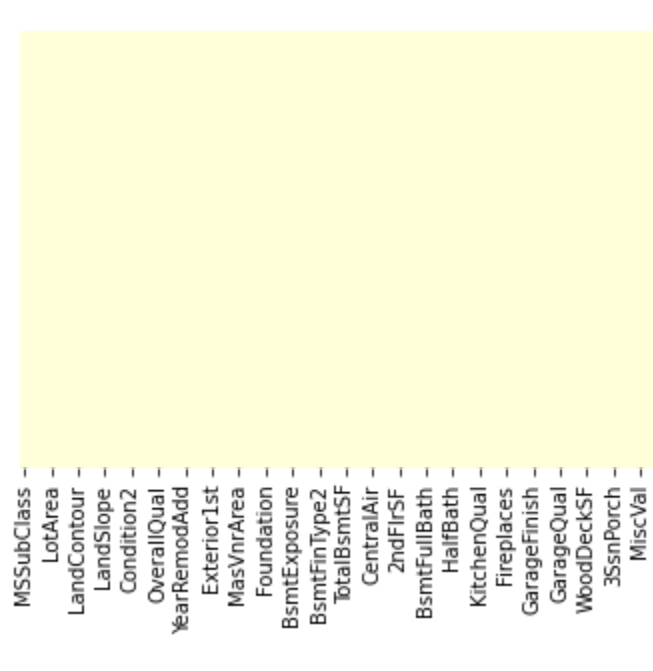
Out[72]: <AxesSubplot:>



In [73]:
```python
house_price['BsmtExposure']=house_price['BsmtExposure'].fillna(house_price['BsmtExposure'].mode()[0])
```

In [74]:
```python
sns.heatmap(house_price.isnull(),yticklabels=False,cbar=False,cmap='YlGnBu')
```

Out[74]: <AxesSubplot:>



In [24]:
```python
house_price['BsmtFinType2']=house_price['BsmtFinType2'].fillna(house_price['BsmtFinType2'].mode()[0])
```

In [25]:
```python
house_price.dropna(inplace=True)
```

In [26]:
```python
house_price.shape
```

Out[26]: (1422, 75)

In [27]:
```python
## Handling categorical feature
columns=['MSZoning', 'LotFrontage', 'LotArea', 'Street','Alley', 'LotShape', 'LandContour', 'Utilities', 'LotConfig','LandSlope','Neighborhood', '
        'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType','MasVnrArea', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual',
        'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1','BsmtFinType2', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating',
        'HeatingQC', 'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF','LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath',
        'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual', 'TotRmsAbvGrd', 'Functional', 'Fireplaces', 'FireplaceQu', 'GarageType',
        'GarageYrBlt', 'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQual','GarageCond', 'PavedDrive', 'WoodDeckSF', 'OpenPorchSF',
        'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'PoolQC','Fence', 'MiscFeature', 'MiscVal', 'MoSold', 'YrSold', 'SaleType',
        'SaleCondition', 'SalePrice']
```

In [28]:
```python
len(columns)
```

Out[28]: 79

In [29]:
```python
def category_multiple(multicolumns):
    df_final=final_df
    i=0
    for fields in multicolumns:
        print(fields)
        df=pd.get_dummies(final_df[fields],drop_first=True)
        final_df.drop([fields],axis=1,inplace=True)
        if i==0:
            df_final=df.copy()
        else:
            df_final=pd.concat([df_final,df],axis=1)
        i=i+1
    df_final=pd.concat([final_df,df_final],axis=1)
    return df_final
```

In [30]:
```python
main_df=house_price.copy()
```

In [31]:
```python
# Combine the test data
test_data=pd.read_csv('formulatest.csv',sep=r'\s*,\s*',header=0, encoding='ascii', engine='python')
```

In [32]:
```python
test_data.shape
```

Out[32]: (93, 76)

In [33]:
```python
test_data.head()
```

Out[33]:

	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	...	OpenPorchSF	EnclosedPorch	3SsnPorch	ScreenPorch	PoolArea
0	160	FV	39.0	3515	Pave	Pave	Reg	Lvl	AllPub	Inside	...	111	0	0	0	0
1	160	FV	30.0	3215	Pave	Pave	Reg	Lvl	AllPub	Inside	...	172	0	0	0	0
2	160	FV	24.0	2544	Pave	Pave	Reg	Lvl	AllPub	Inside	...	172	0	0	0	0
3	160	FV	24.0	2544	Pave	Pave	Reg	Lvl	AllPub	Inside	...	166	0	0	0	0
4	20	FV	57.0	12853	Pave	Pave	IR1	Lvl	AllPub	Inside	...	192	0	224	0	0

5 rows × 76 columns

In [34]:
```python
final_data=pd.concat([house_price,test_data],axis=0)
```

In [35]:
```python
print(final_data.columns)
```

```
Index(['MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Street',
       'LotShape', 'LandContour', 'Utilities', 'LotConfig', 'LandSlope',
       'Neighborhood', 'Condition1', 'Condition2', 'BldgType', 'HouseStyle',
       'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd', 'RoofStyle',
       'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType', 'MasVnrArea',
       'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual', 'BsmtCond',
       'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1', 'BsmtFinType2',
       'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating', 'HeatingQC',
       'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF',
       'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath', 'HalfBath',
       'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual', 'TotRmsAbvGrd',
       'Functional', 'Fireplaces', 'FireplaceQu', 'GarageType', 'GarageFinish',
       'GarageCars', 'GarageArea', 'GarageQual', 'GarageCond', 'PavedDrive',
       'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch',
       'ScreenPorch', 'PoolArea', 'MiscVal', 'MoSold', 'YrSold', 'SaleType',
       'SaleCondition', 'SalePrice', 'Alley', 'GarageYrBlt'],
      dtype='object')
```

In [36]:
```python
final_data.shape
```

Out[36]: (1515, 77)

In [37]:
```python
data_train=final_data.iloc[:1400,:]
data_test=final_data.iloc[1400:,:]
```

In [38]:
```python
data_test.drop(['SalePrice'],axis=1,inplace=True)
```

In [39]:
```python
data_test.shape
```

Out[39]: (115, 76)

In [58]:
```python
## Predection of data set using algoritham
x=data_train[['SalePrice']]
y=data_train['SalePrice']
```

In [59]:
```python
X_train,X_test,Y_train,Y_test = train_test_split(x,y,test_size=0.1,random_state=1)
```

In [60]:
```python
print(X_train)
print(X_test)
```

```
      SalePrice
378    394432.0
638     85000.0
1425   142000.0
1358   177500.0
174    184000.0
...         ...
732    222500.0
928    236500.0
1127   259000.0
241    110500.0
1092   136500.0

[1260 rows x 1 columns]
      SalePrice
562    108000.0
773    114500.0
1318   275000.0
1159   185000.0
1094   129000.0
...         ...
594    110000.0
1407   112000.0
471    190000.0
916     35311.0
1161   224000.0

[140 rows x 1 columns]
```

In [43]:
```python
# Training the model
le =LinearRegression()
#classifier=LinearRegression()
```

In [44]:
```python
#Training model with fit line
le.fit(x_train,y_train)
```

Out[44]:

▾ LinearRegression
LinearRegression()

In [53]:
```python
le.intercept_
```

Out[53]: 193578.75021020928

In [54]:
```python
le.coef_
```

Out[54]: array([-180.87121729])

In [61]:
```python
##Accuracy of testing data
le.score(X_test,Y_test)
y_predection=le.predict(X_test)
```

In [66]:
```python
print(r2_score(y_predection, X_test))
```

-4.421377730775149

In [50]:
```python
import pandas as pd
data=pd.read_csv('train.csv')
data
```

Out[50]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	YrSold	SaleType
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2008	WD
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	5	2007	WD
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	9	2008	WD
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	2	2006	WD
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	12	2008	WD
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1455	1456	60	RL	62.0	7917	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	8	2007	WD
1456	1457	20	RL	85.0	13175	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	MnPrv	NaN	0	2	2010	WD
1457	1458	70	RL	66.0	9042	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	GdPrv	Shed	2500	5	2010	WD
1458	1459	20	RL	68.0	9717	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	4	2010	WD
1459	1460	20	RL	75.0	9937	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	NaN	0	6	2008	WD

1460 rows × 81 columns

In [51]:
```python
data['BedroomAbvGr'].value_counts()
```

Out[51]:
```
3    804
2    358
4    213
1     50
5     21
6      7
0      6
8      1
Name: BedroomAbvGr, dtype: int64
```

In [52]:
```python
data['FullBath'].value_counts()
```

Out[52]:
```
2    768
1    650
3     33
0      9
Name: FullBath, dtype: int64
```

In [ ]: