Project on House Predication Data

In [1]:
```
from warnings import filterwarnings
filterwarnings('ignore')
```

Step 1 : Read Training CSV file

In [2]:
```
import pandas as pd
df = pd.read_csv('training_set.csv')
pd.set_option('display.max_columns',None)
df.head()
```

Out[2]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	CollgCr
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	FR2	Gtl	Veenker
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	CollgCr
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	Crawfor
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	FR2	Gtl	NoRidge

step 2 : Check Data Quality and missing values

In [3]:
```
df.shape
```

Out[3]:
```
(1460, 81)
```

In [4]:
```
df.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     588 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]:
```
df.isna().sum()
```

Out[5]:
```
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 [6]:
```
mis = df.isna().sum()
mis[mis>=1]
```

Out[6]:
```
LotFrontage     259
Alley          1369
MasVnrType      872
MasVnrArea        8
BsmtQual         37
BsmtCond         37
BsmtExposure     38
BsmtFinType1     37
BsmtFinType2     38
Electrical        1
FireplaceQu     690
GarageType       81
GarageYrBlt      81
GarageFinish     81
GarageQual       81
GarageCond       81
PoolQC         1453
Fence          1179
MiscFeature    1406
dtype: int64
```

In [7]:
```
df.duplicated().sum()
```

Out[7]:
```
0
```

In [8]:
```
df.columns
```

Out[8]:
```
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')
```

step 3 : Perform EDA on categorical and contineous features

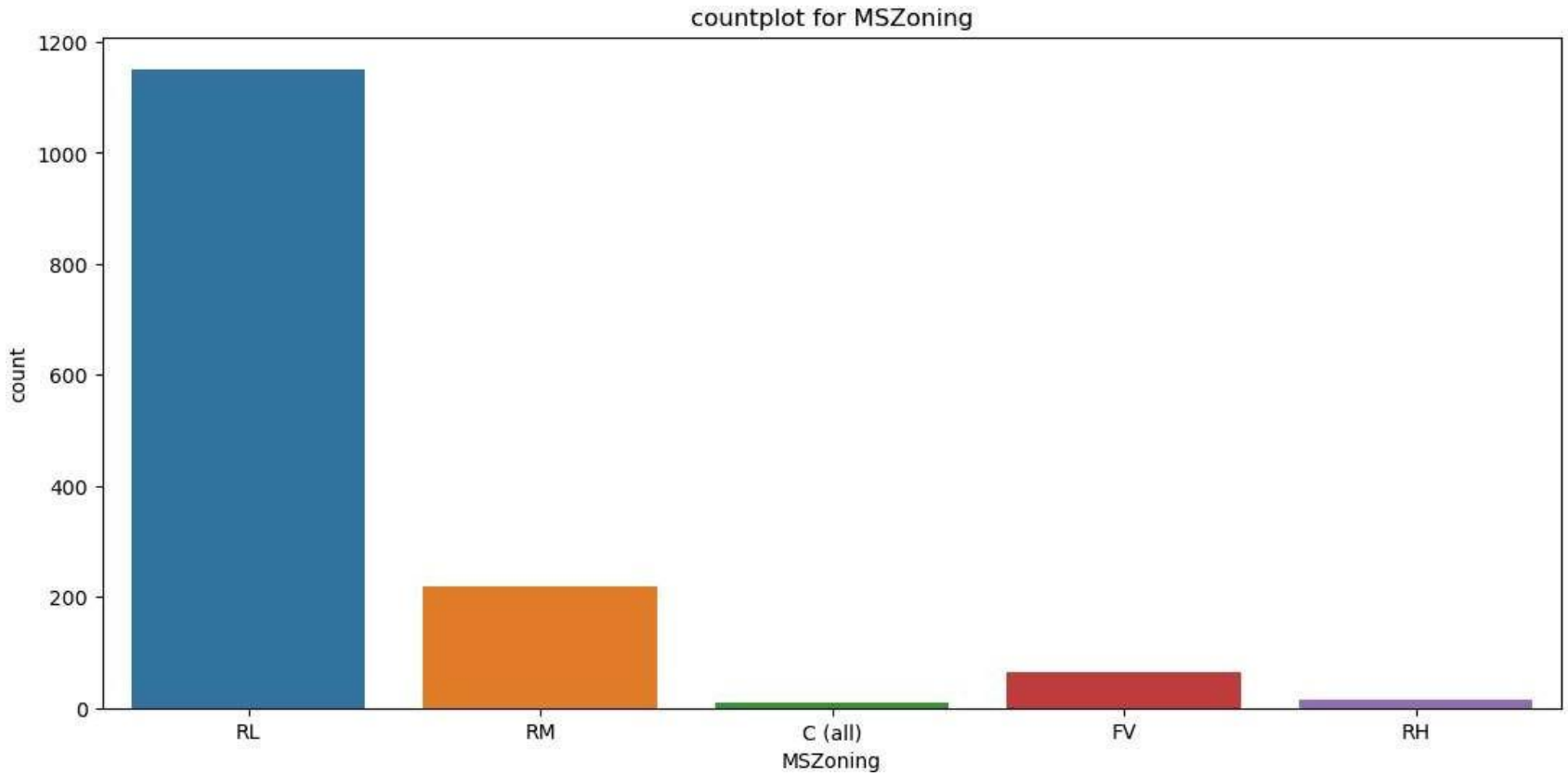
1. Univariate
2. Bivariate

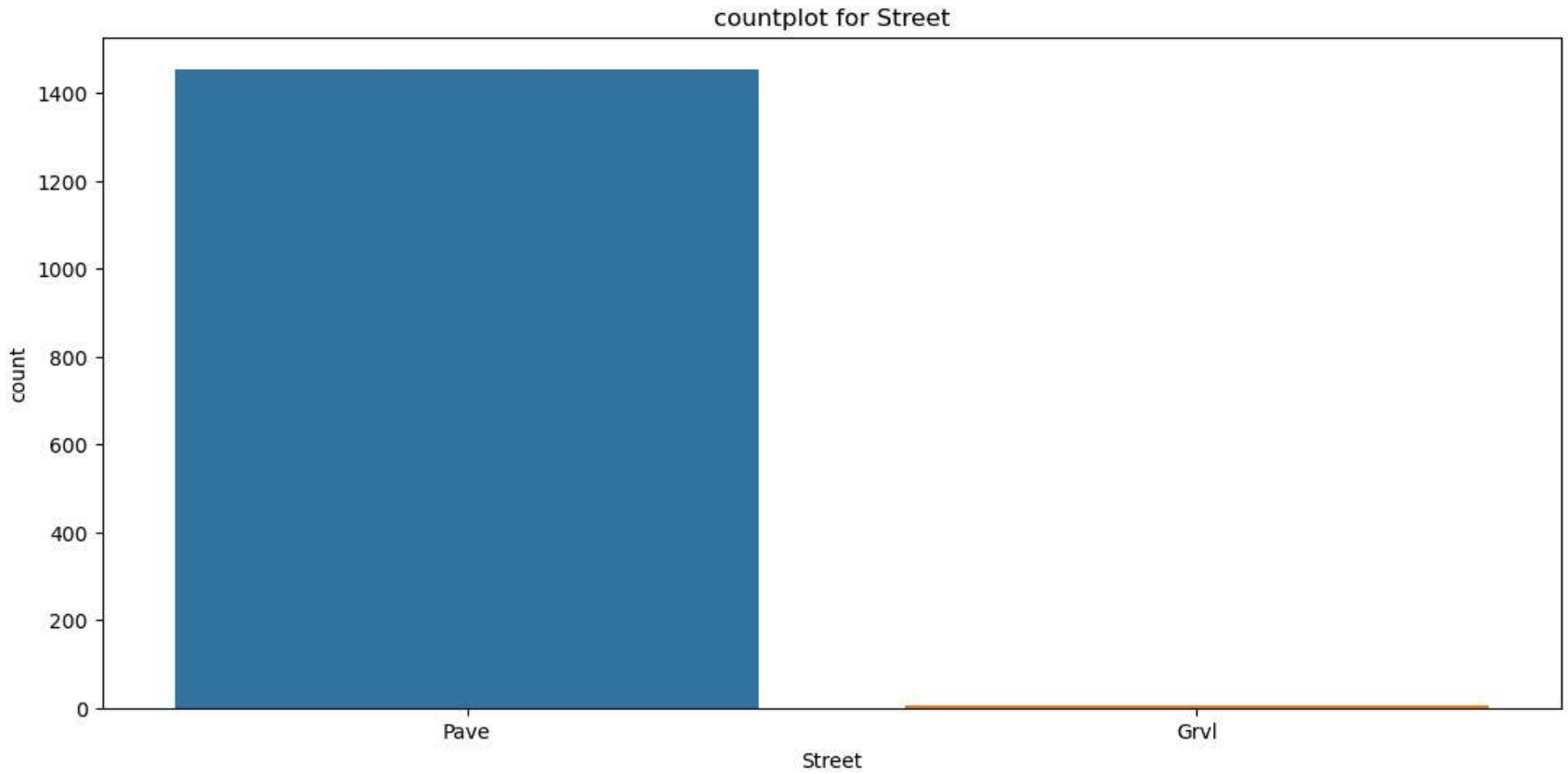
1. Univarate analysis

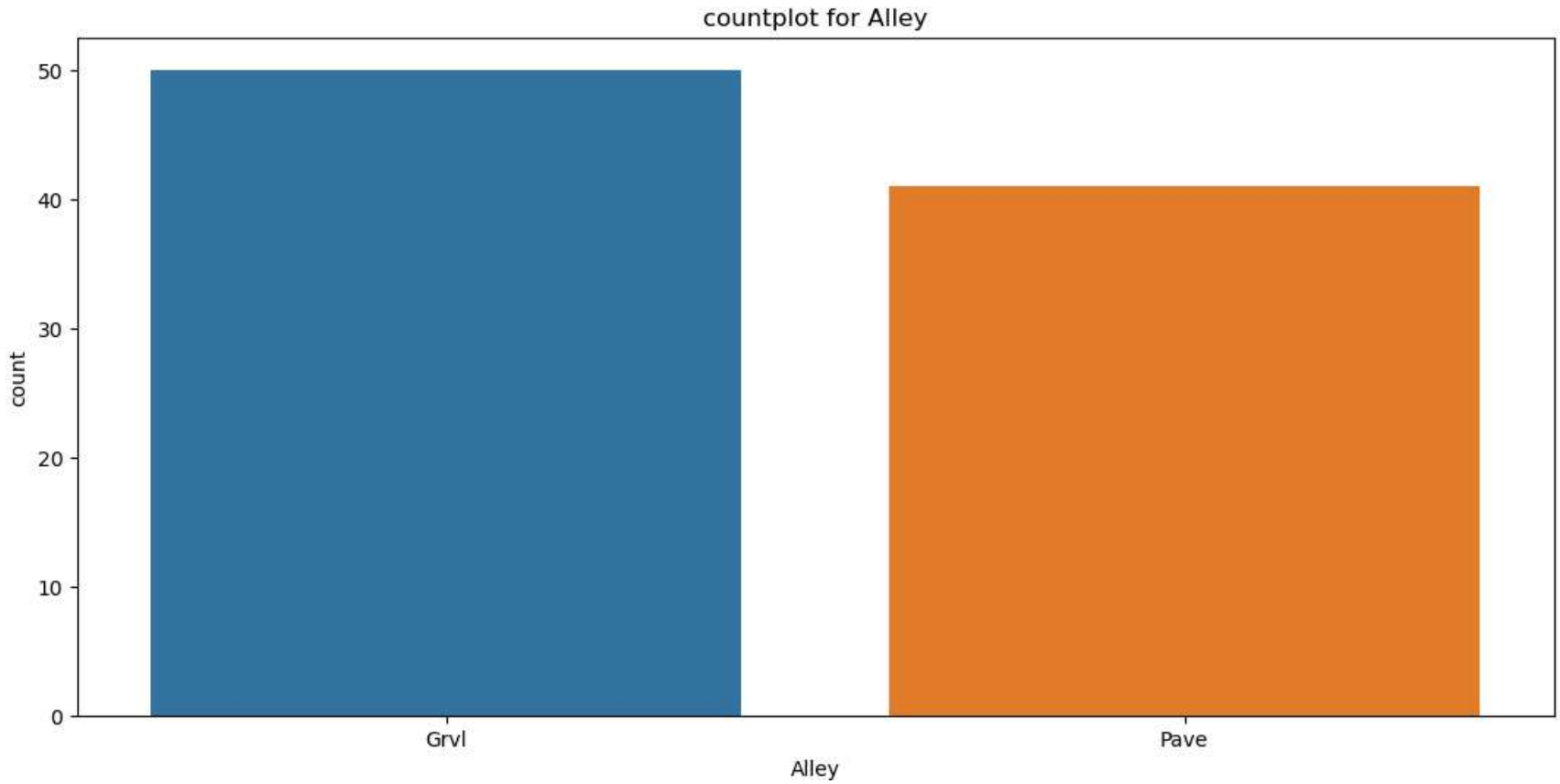
In [9]:
```
# Here i am using my userdefined function for univariate data analysis
from oops import univariate
```

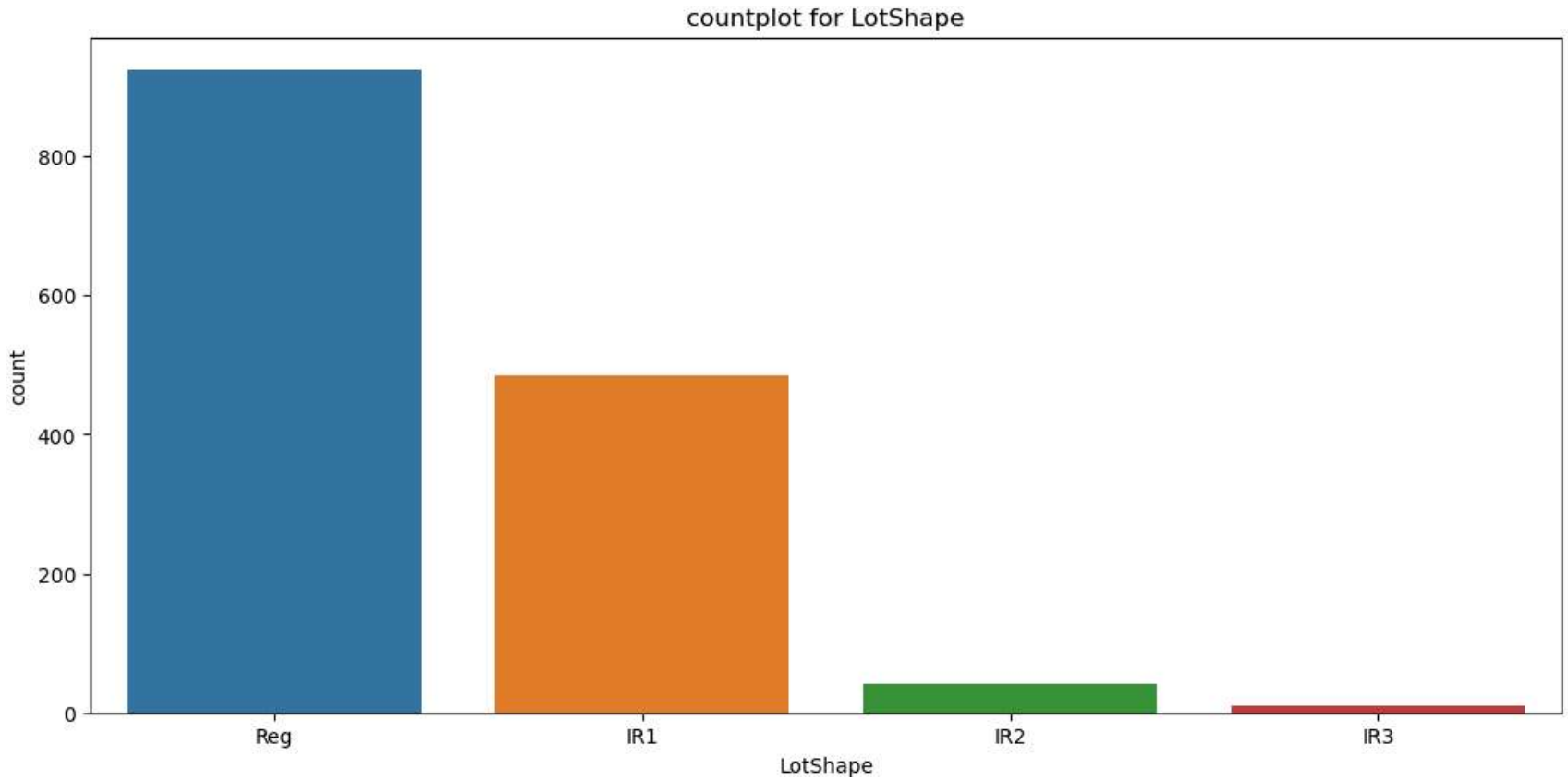
In [10]:
```
univariate(df)
```

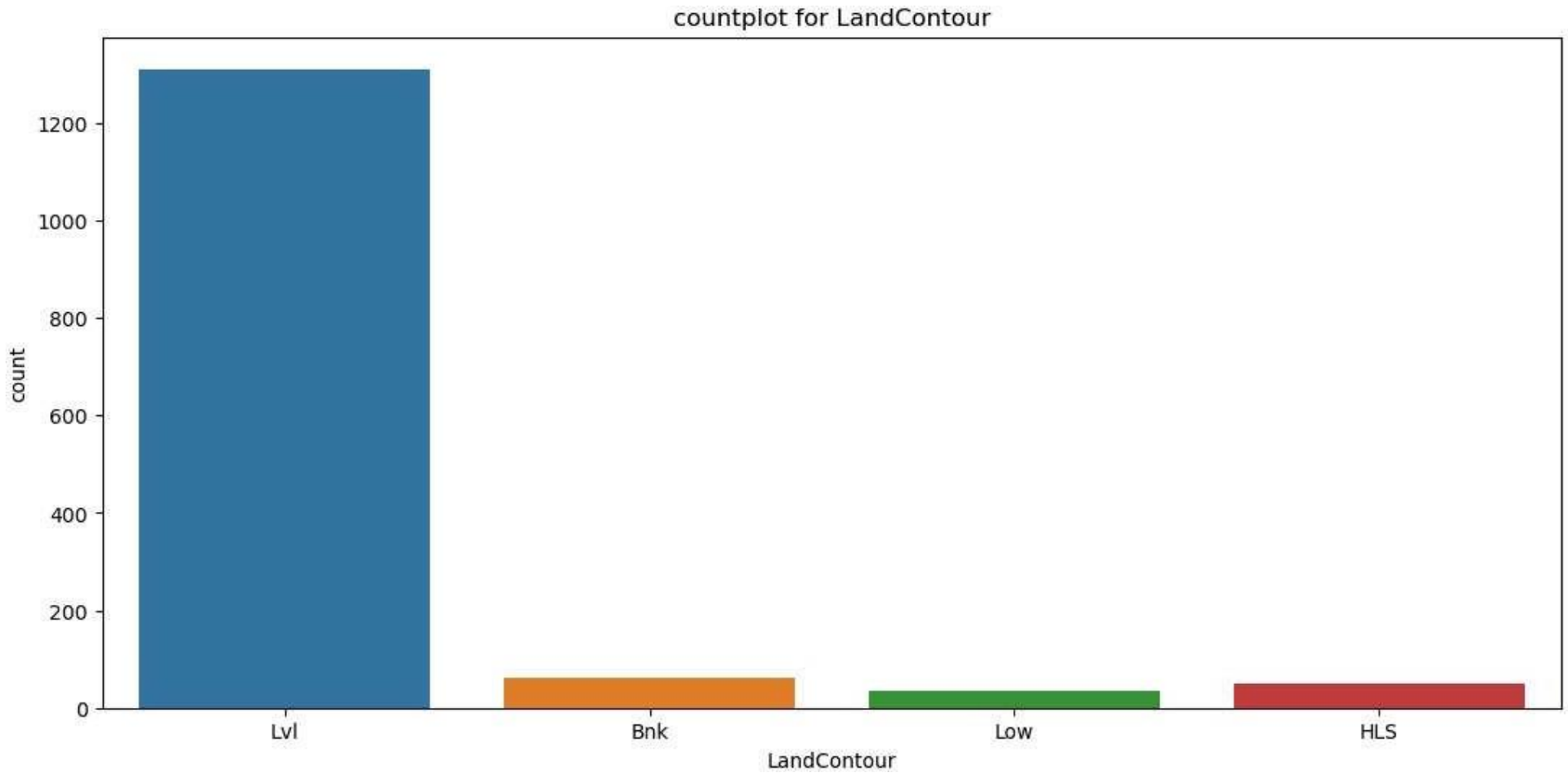
```
Categorical variable : ['MSZoning', 'Street', 'Alley', 'LotShape', 'LandContour', 'Utilities', 'LotConfig', 'LandSlop
e', 'Neighborhood', 'Condition1', 'Condition2', 'BldgType', 'HouseStyle', 'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exte
rior2nd', 'MasVnrType', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1',
'BsmtFinType2', 'Heating', 'HeatingQC', 'CentralAir', 'Electrical', 'KitchenQual', 'Functional', 'FireplaceQu', 'Garage
Type', 'GarageFinish', 'GarageQual', 'GarageCond', 'PavedDrive', 'PoolQC', 'Fence', 'MiscFeature', 'SaleType', 'SaleCon
dition'] countplot
```

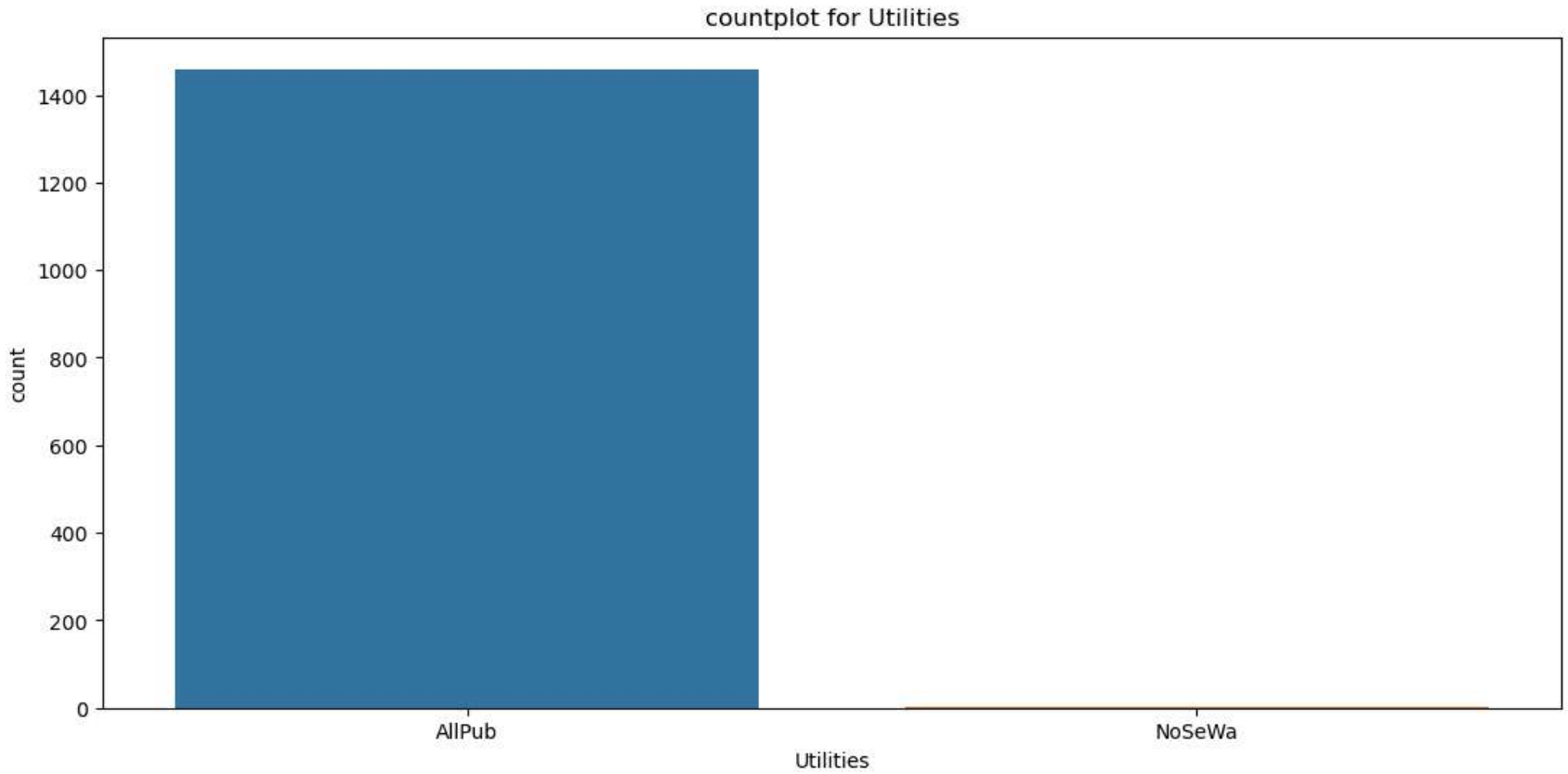

countplot for MSZoning
count
1200
1000
800
600
400
200
0
RL
RM
C (all)
FV
RH
MSZoning

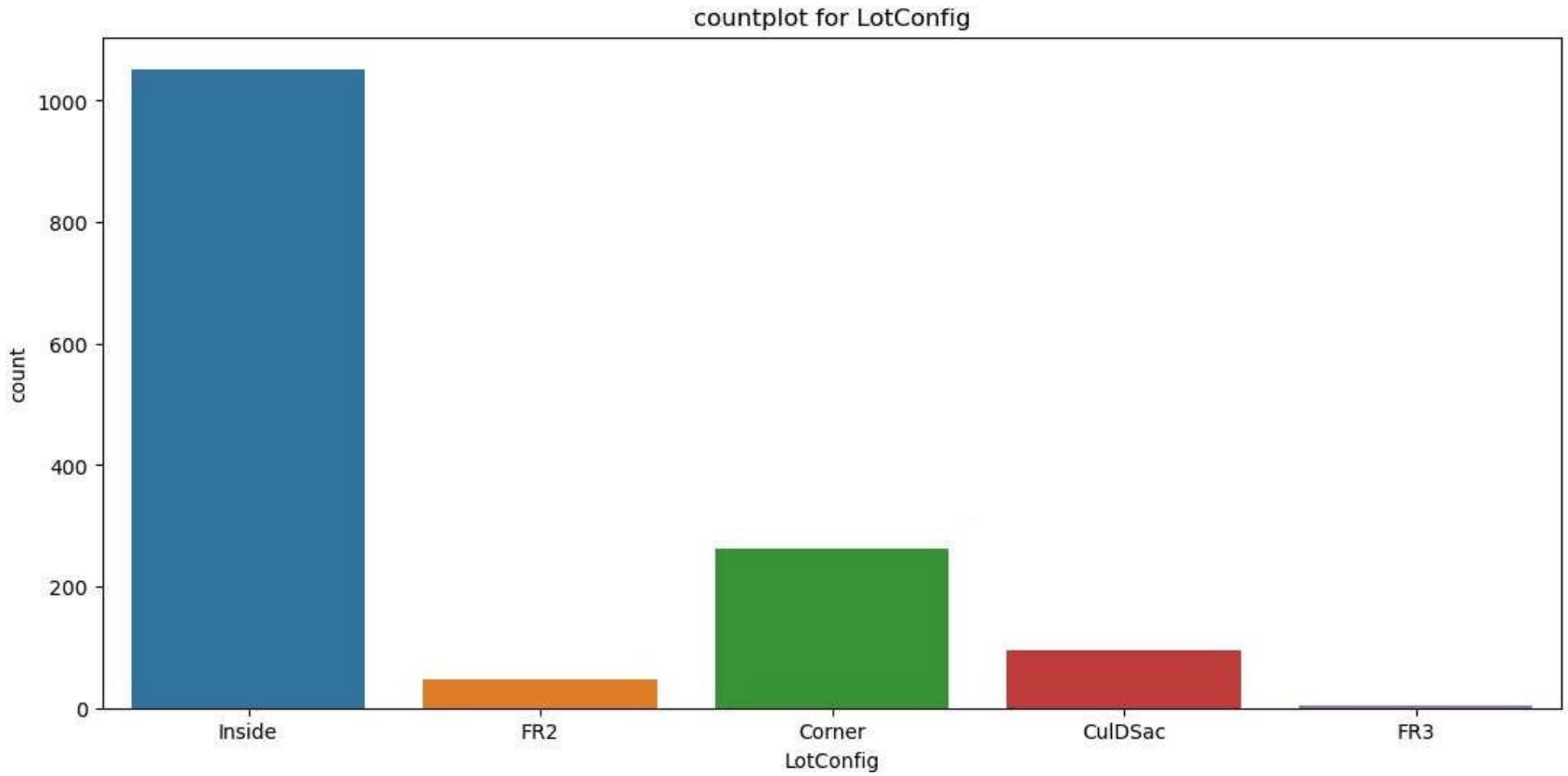

countplot for Street
count
1400
1200
1000
800
600
400
200
0
Pave
Grvl
Street

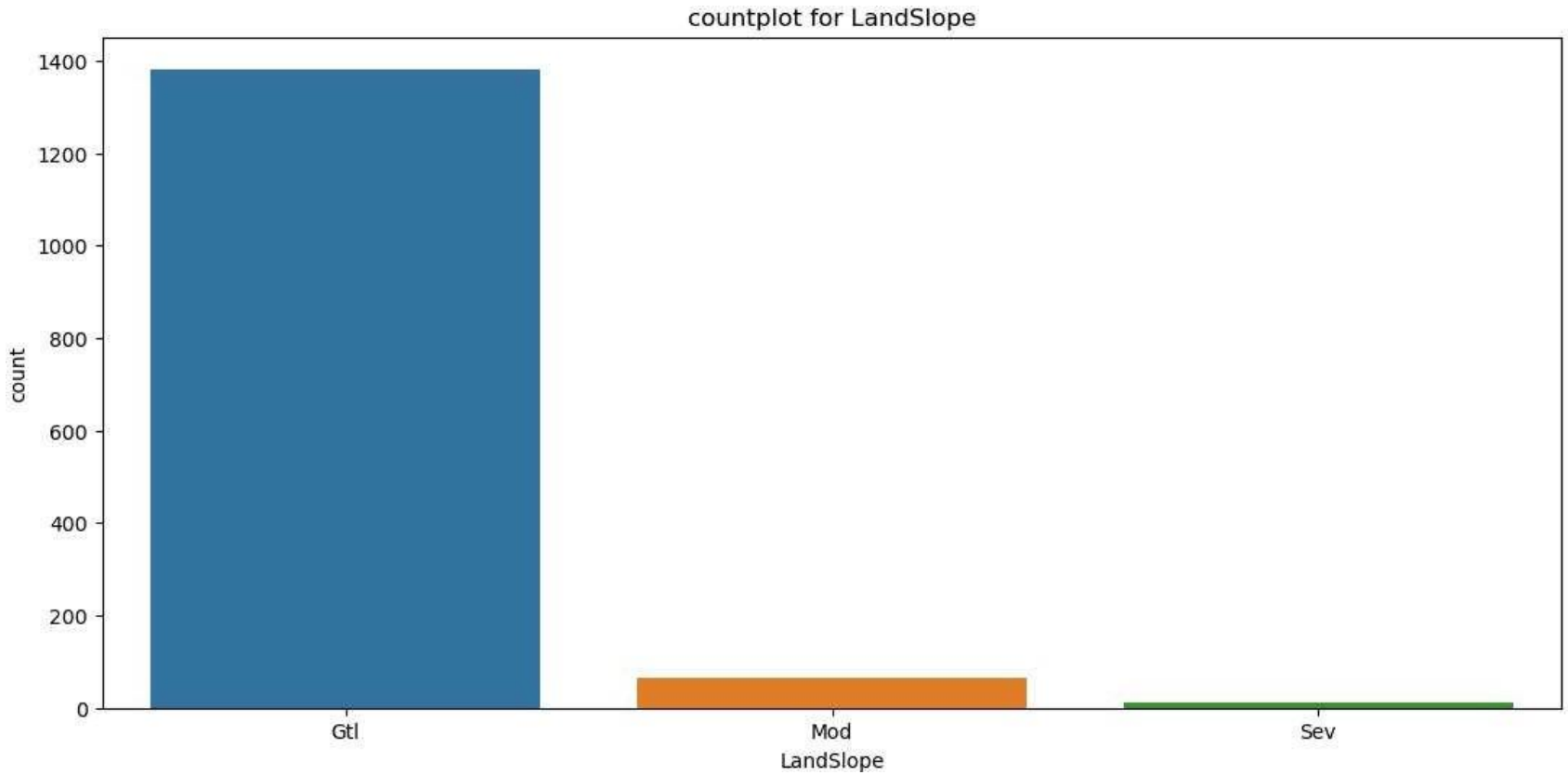

countplot for Alley
50
40
30
20
10
0
count
Grvl
Pave
Alley

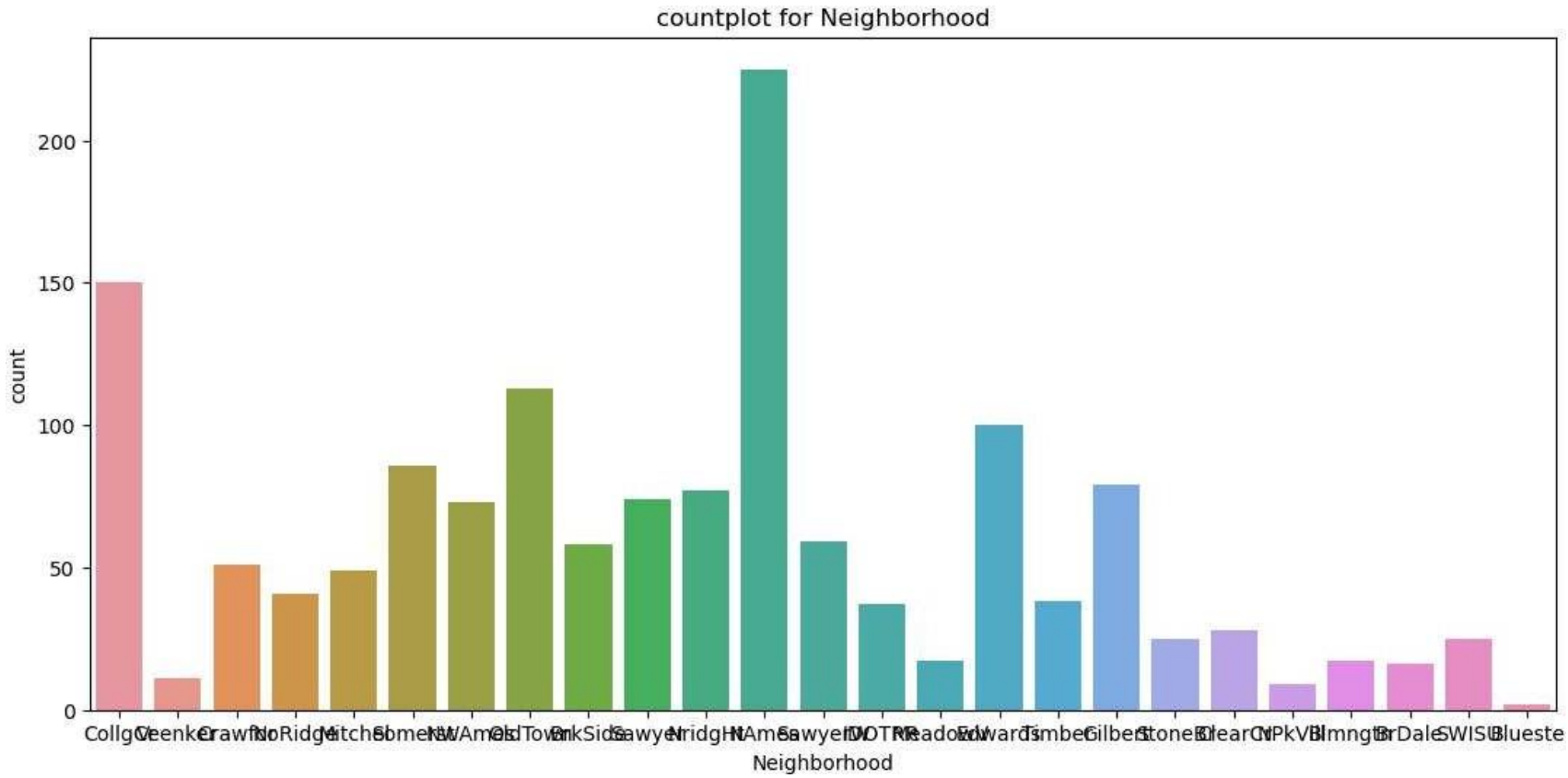

countplot for LotShape
count
800
600
400
200
0
Reg
IR1
IR2
IR3
LotShape

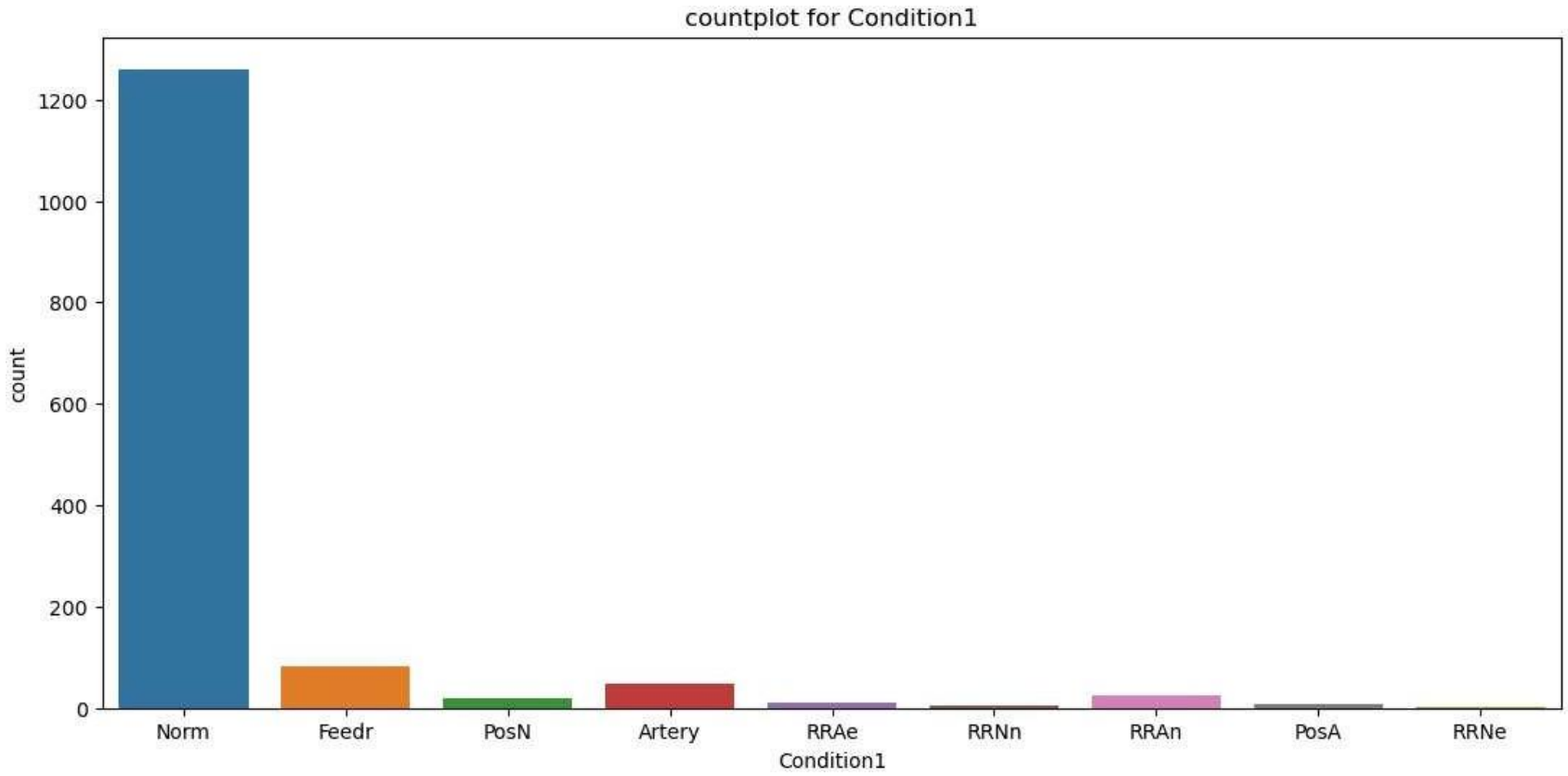

countplot for LandContour
count
1200
1000
800
600
400
200
0
Lvl
Bnk
Low
HLS
LandContour

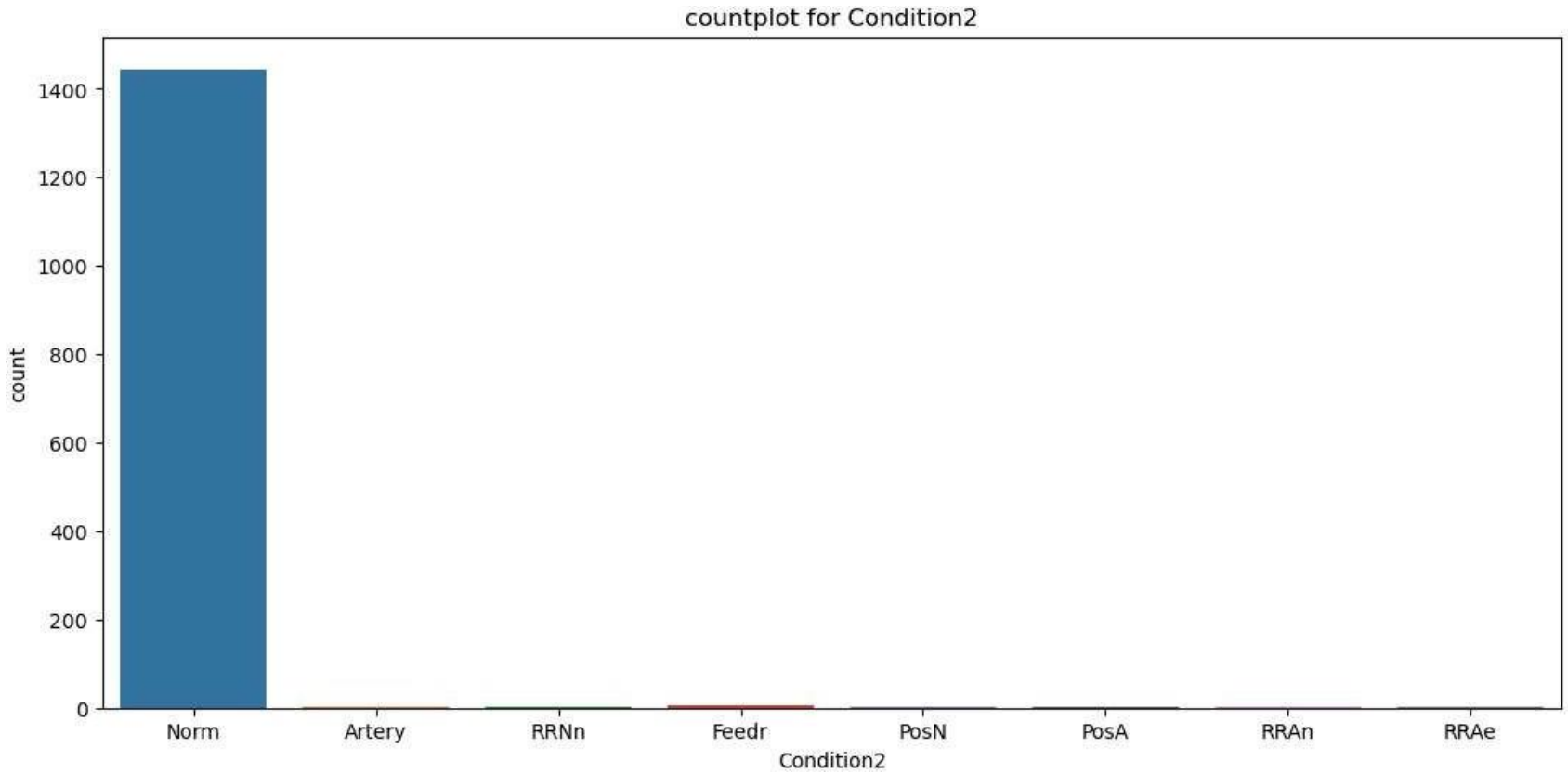

countplot for Utilities
count
1400
1200
1000
800
600
400
200
0
AllPub
NoSeWa
Utilities

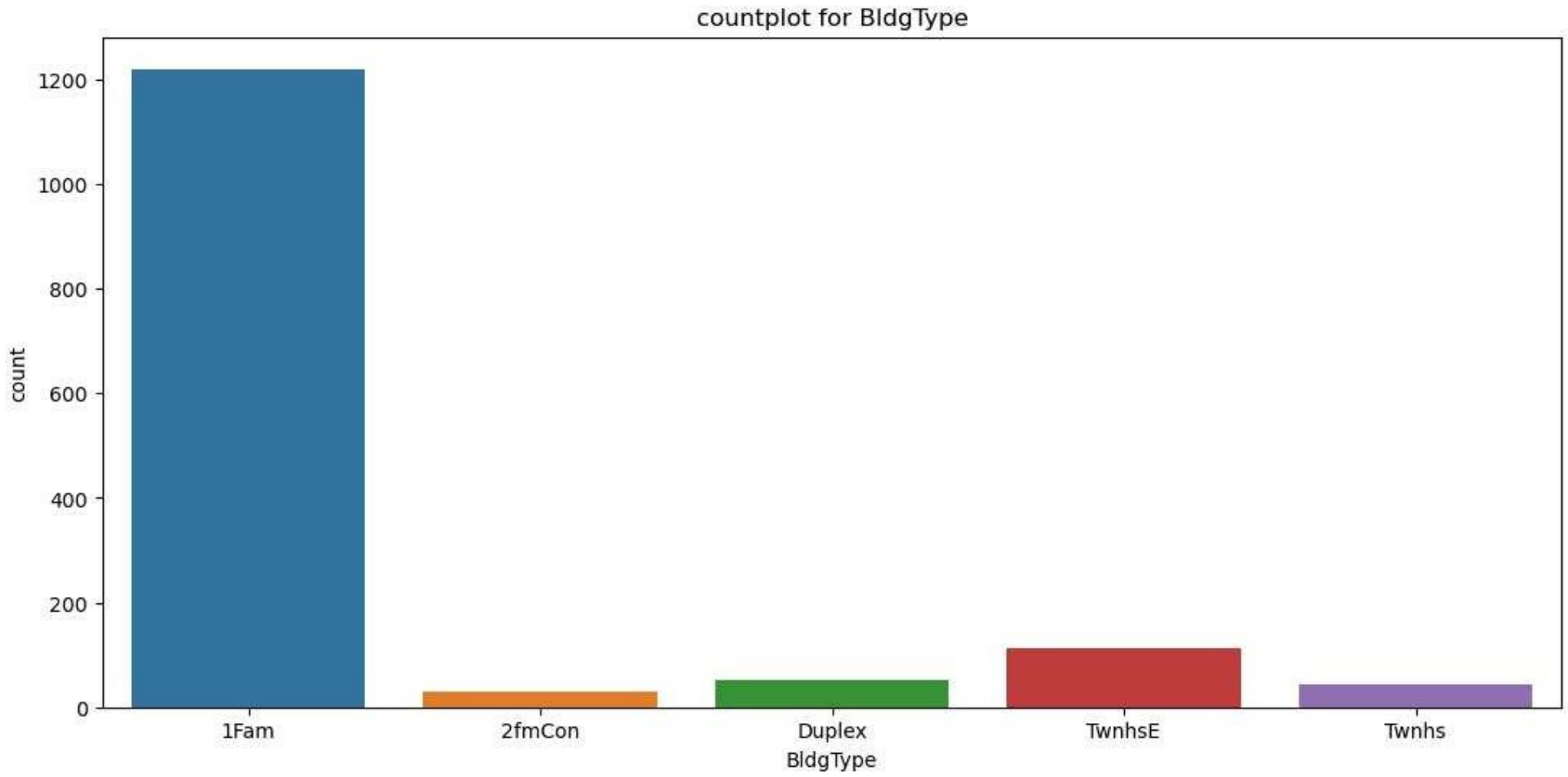

countplot for LotConfig
count
1000
800
600
400
200
0
Inside
FR2
Corner
CulDSac
FR3
LotConfig

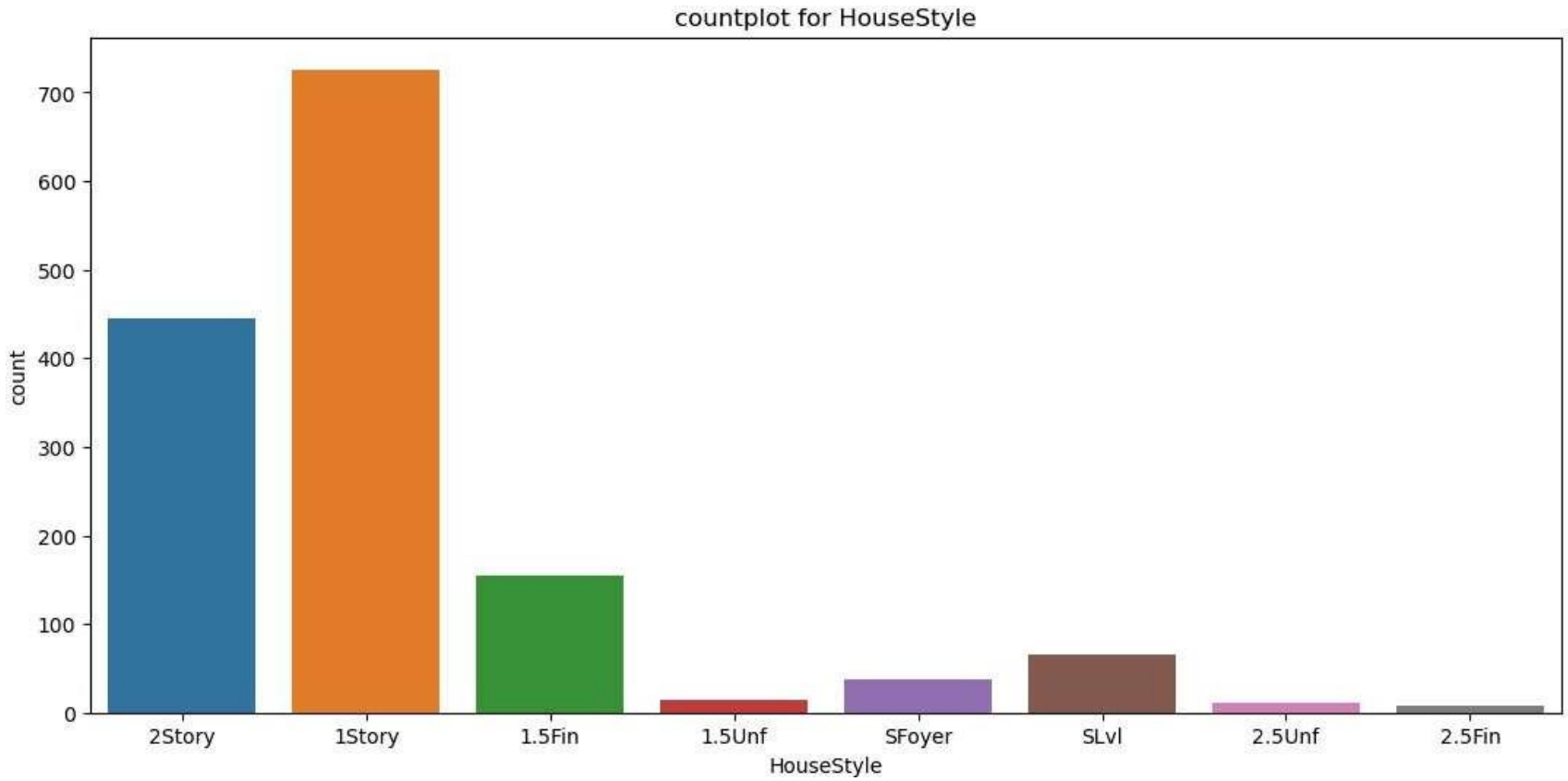

countplot for LandSlope
1400
1200
1000
800
600
400
200
0
count
Gtl
Mod
Sev
LandSlope

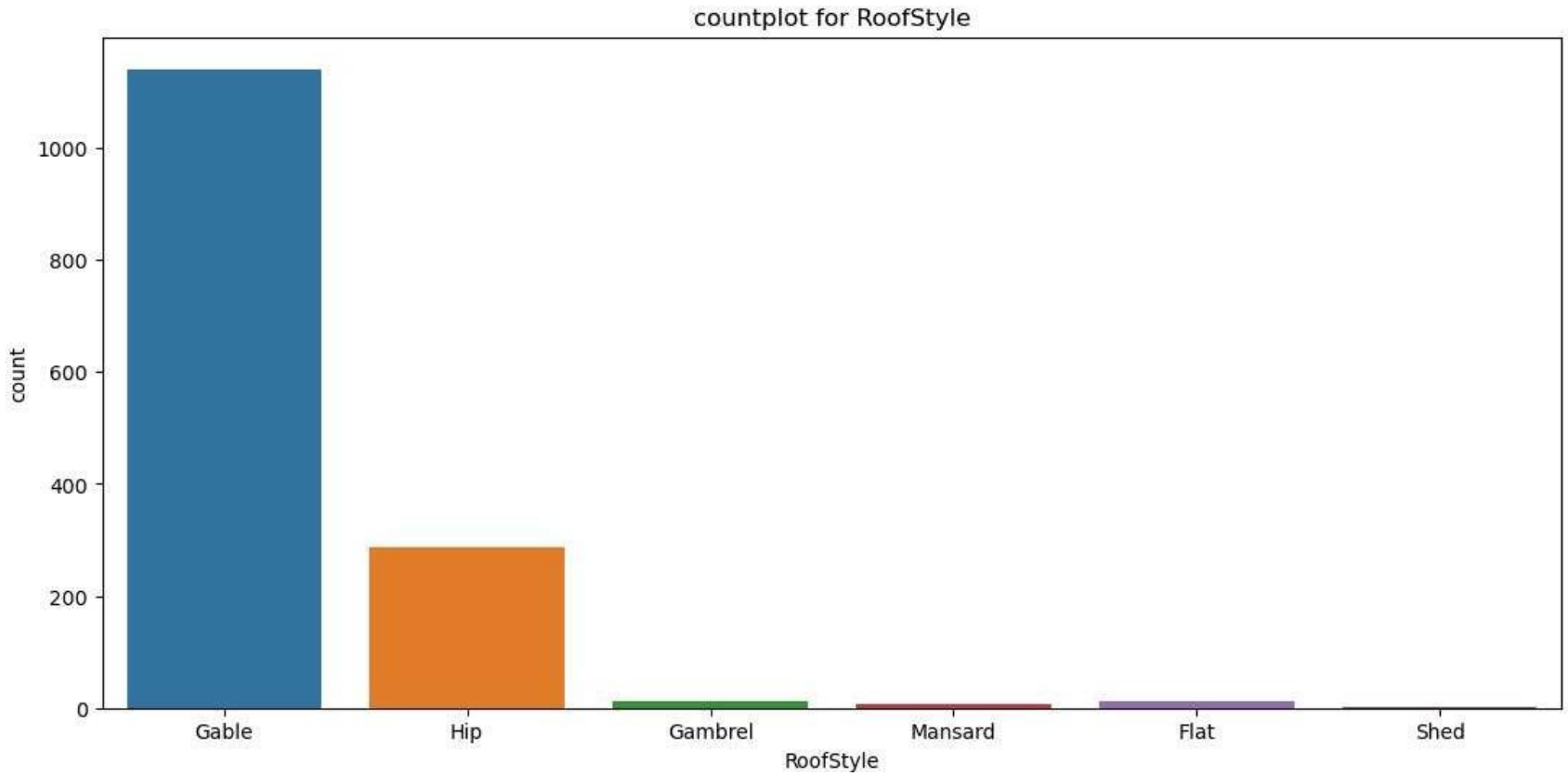

countplot for Neighborhood
count
0
50
100
150
200
Neighborhood

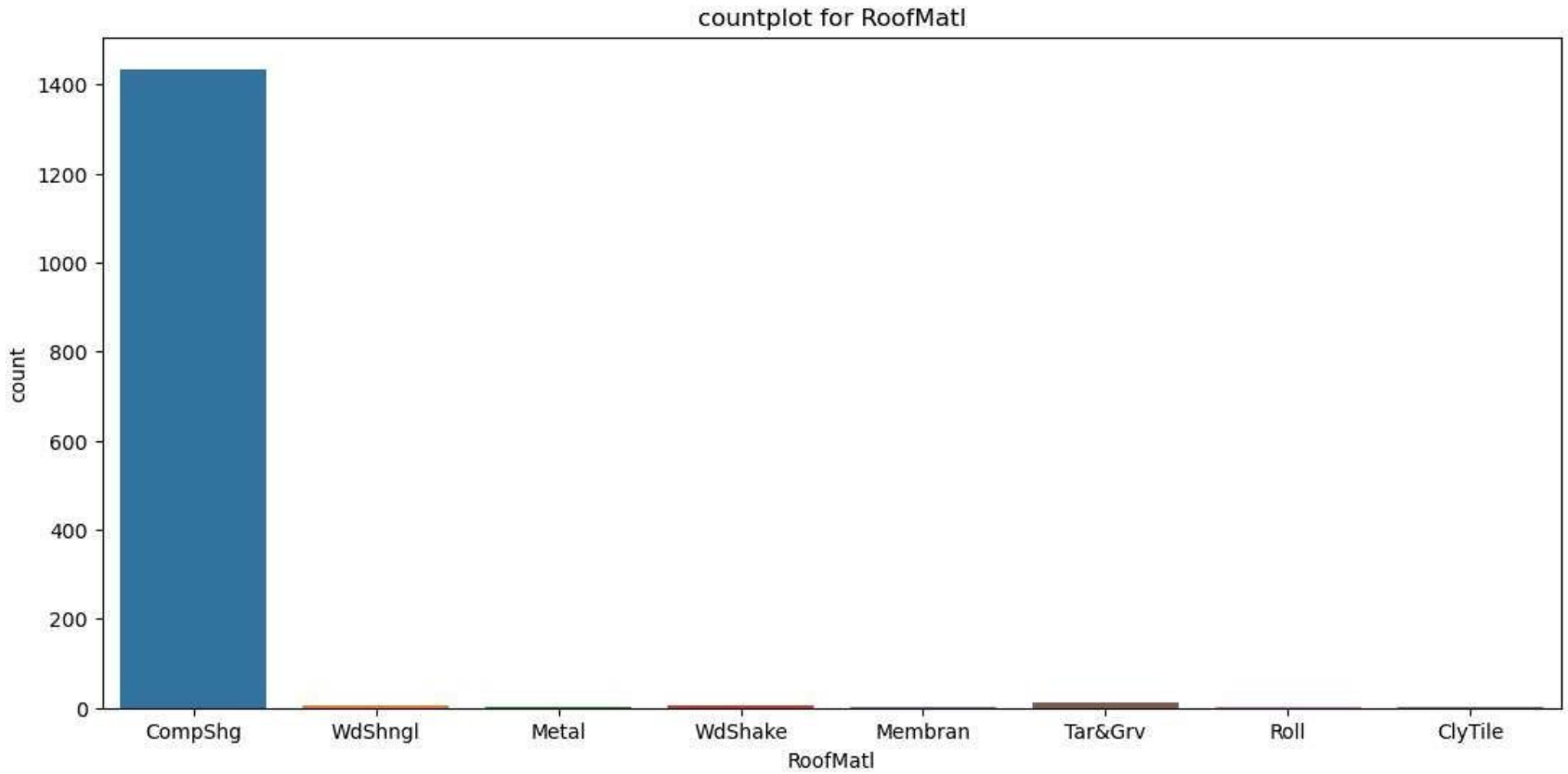

countplot for Condition1
count
1200
1000
800
600
400
200
0
Norm
Feedr
PosN
Artery
RRAe
RRNn
RRAn
PosA
RRNe
Condition1

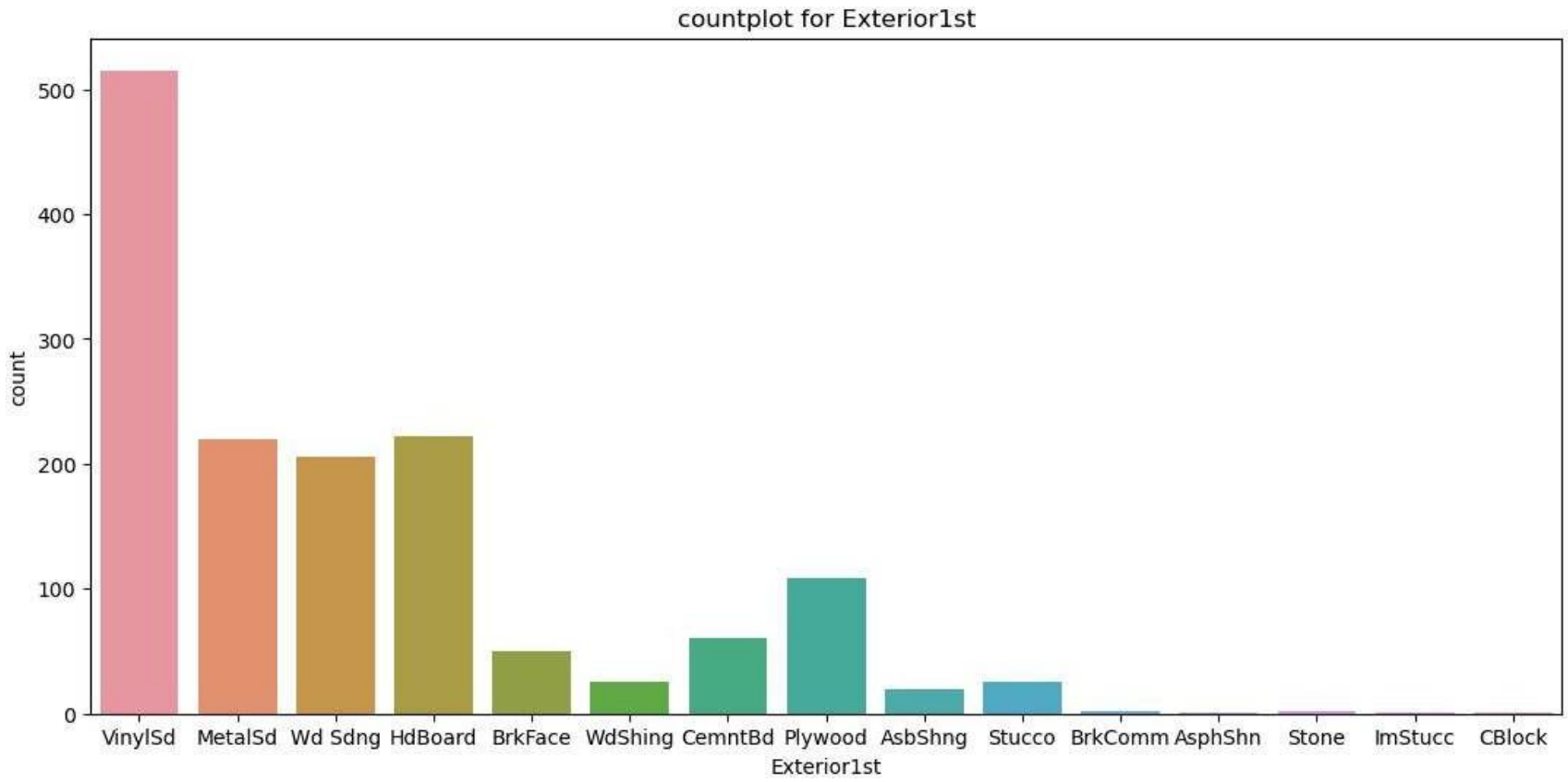

countplot for Condition2
count
1400
1200
1000
800
600
400
200
0
Norm
Artery
RRNn
Feedr
PosN
PosA
RRAn
RRAe
Condition2

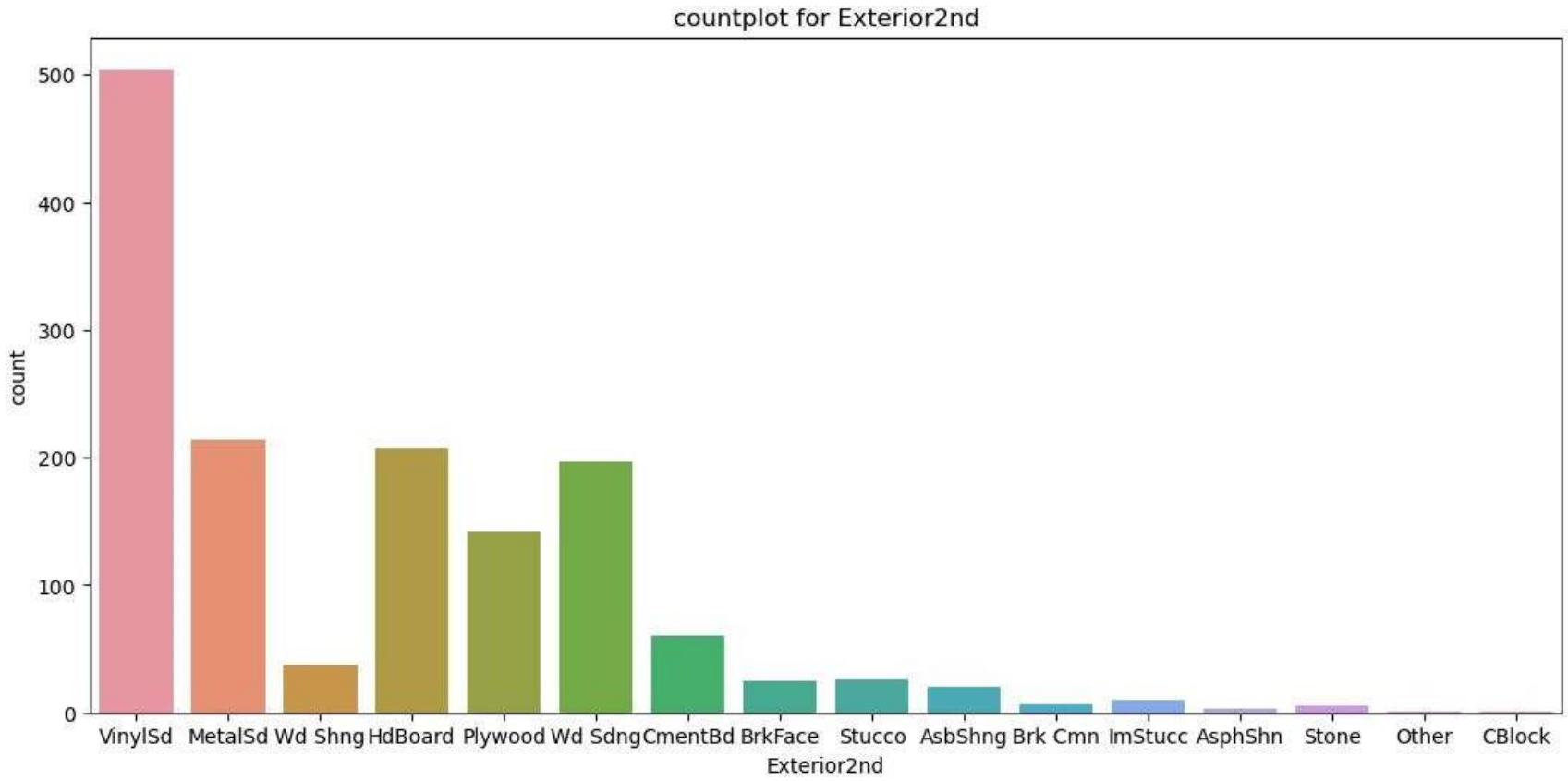

countplot for BldgType
count
1200
1000
800
600
400
200
0
1Fam
2fmCon
Duplex
TwnhsE
Twnhs
BldgType

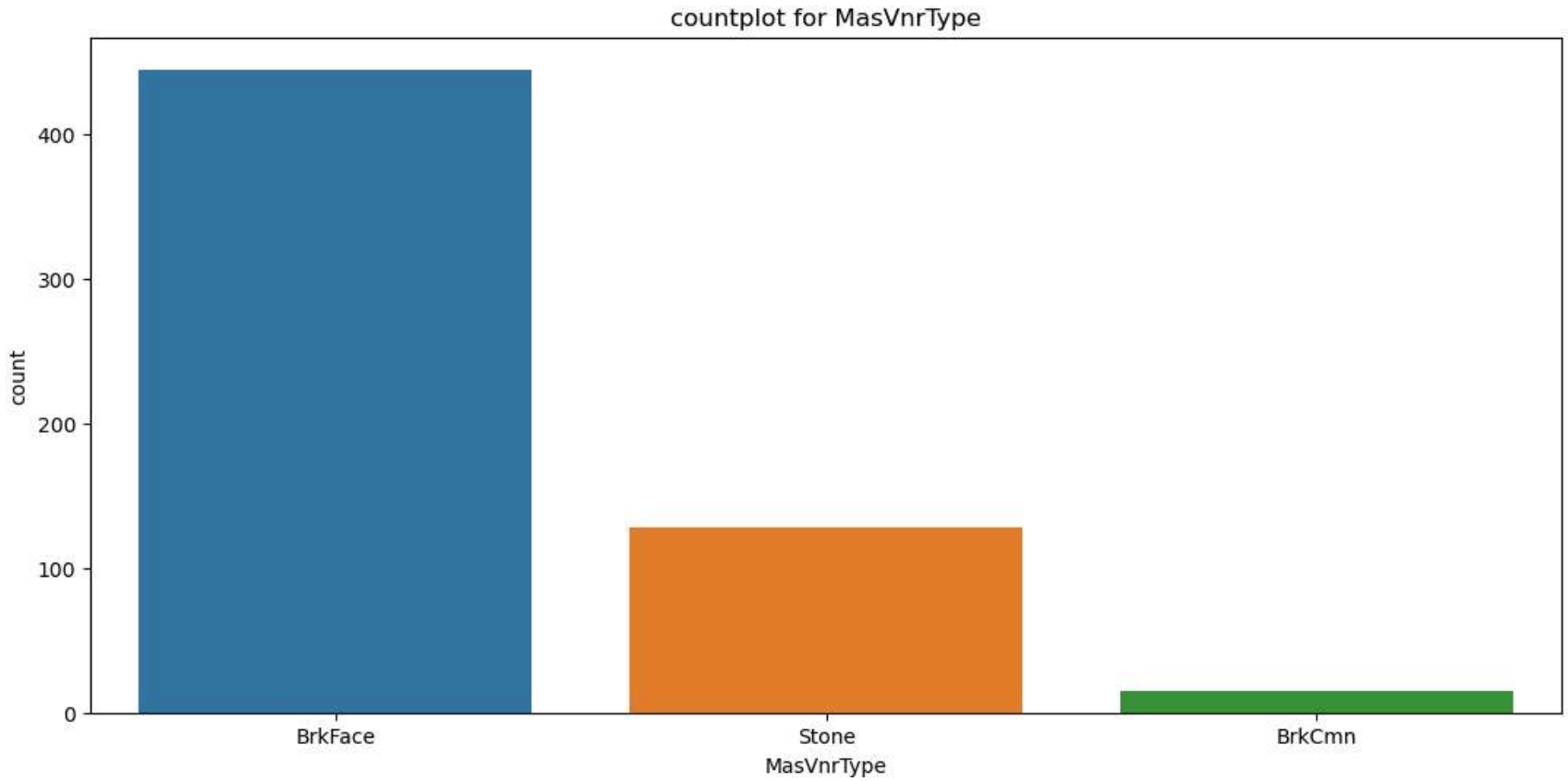

countplot for HouseStyle
count
0
100
200
300
400
500
600
700
2Story
1Story
1.5Fin
1.5Unf
SFoyer
SLvl
2.5Unf
2.5Fin
HouseStyle

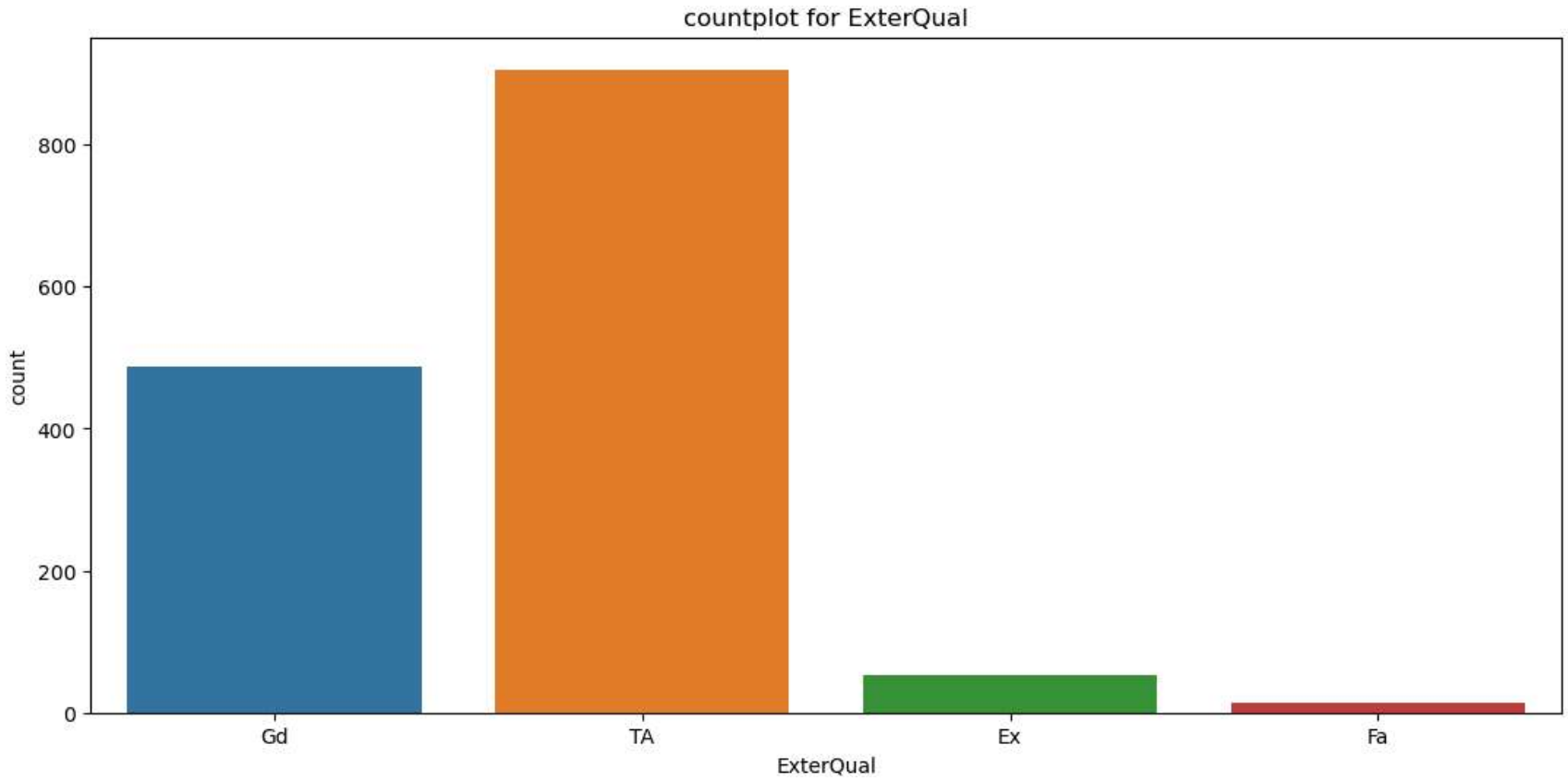

countplot for RoofStyle
count
1000
800
600
400
200
0
Gable
Hip
Gambrel
Mansard
Flat
Shed
RoofStyle

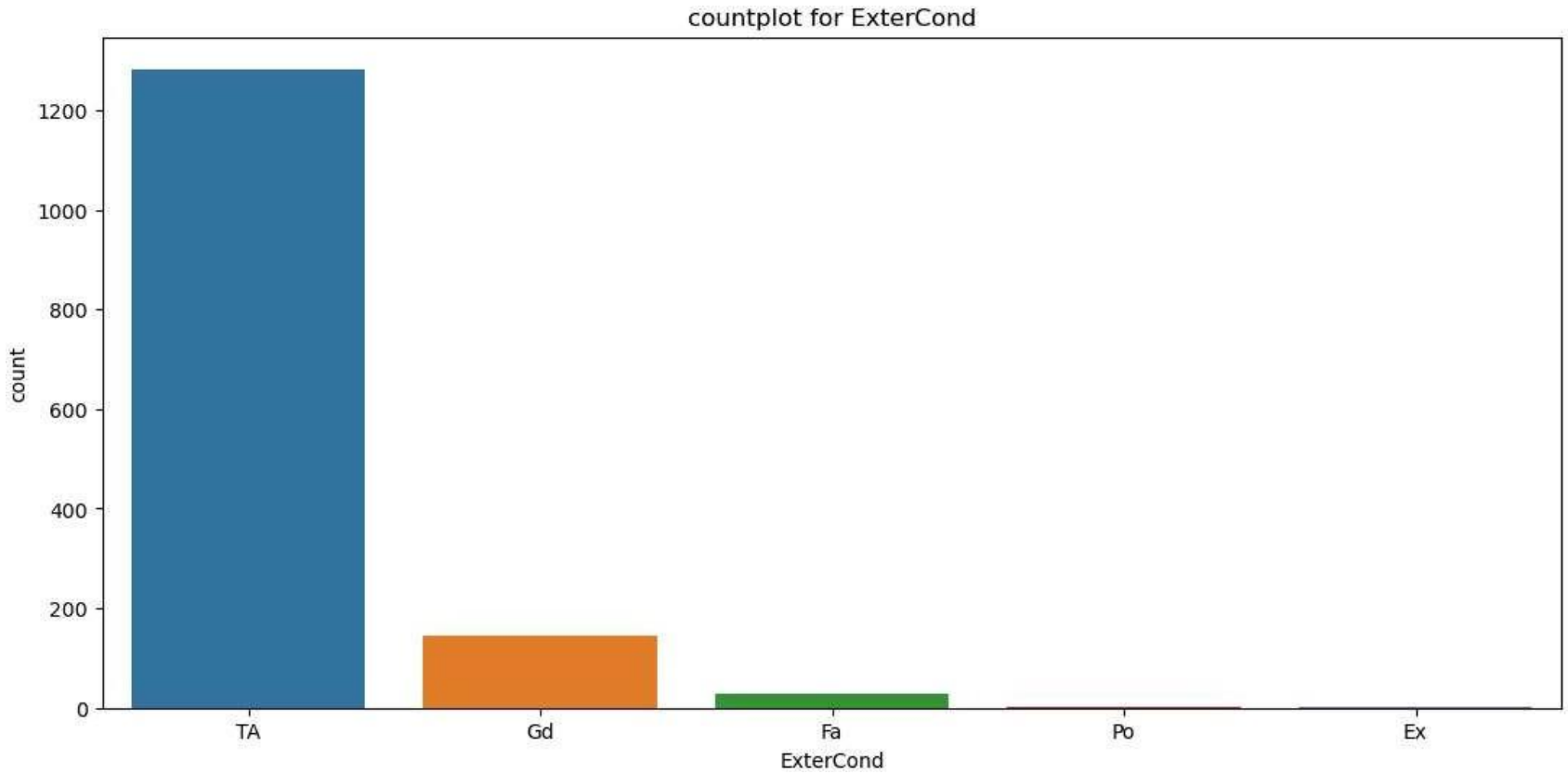

countplot for RoofMatl
count
1400
1200
1000
800
600
400
200
0
CompShg
WdShngl
Metal
WdShake
Membran
Tar&Grv
Roll
ClyTile
RoofMatl

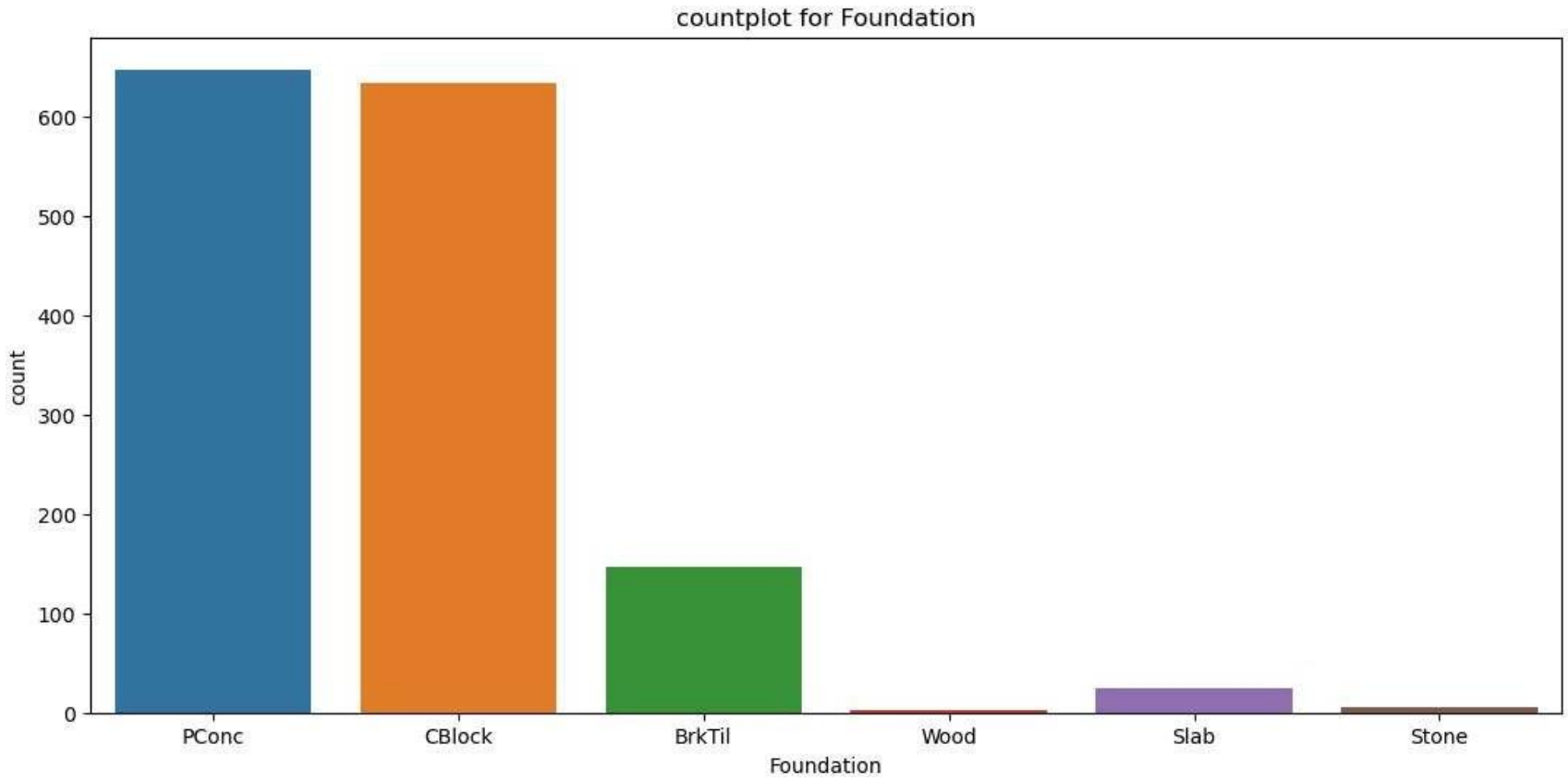

countplot for Exterior1st
count
0
100
200
300
400
500
VinylSd
MetalSd
Wd Sdng
HdBoard
BrkFace
WdShing
CemntBd
Plywood
AsbShng
Stucco
BrkComm
AsphShn
Stone
ImStucc
CBlock
Exterior1st

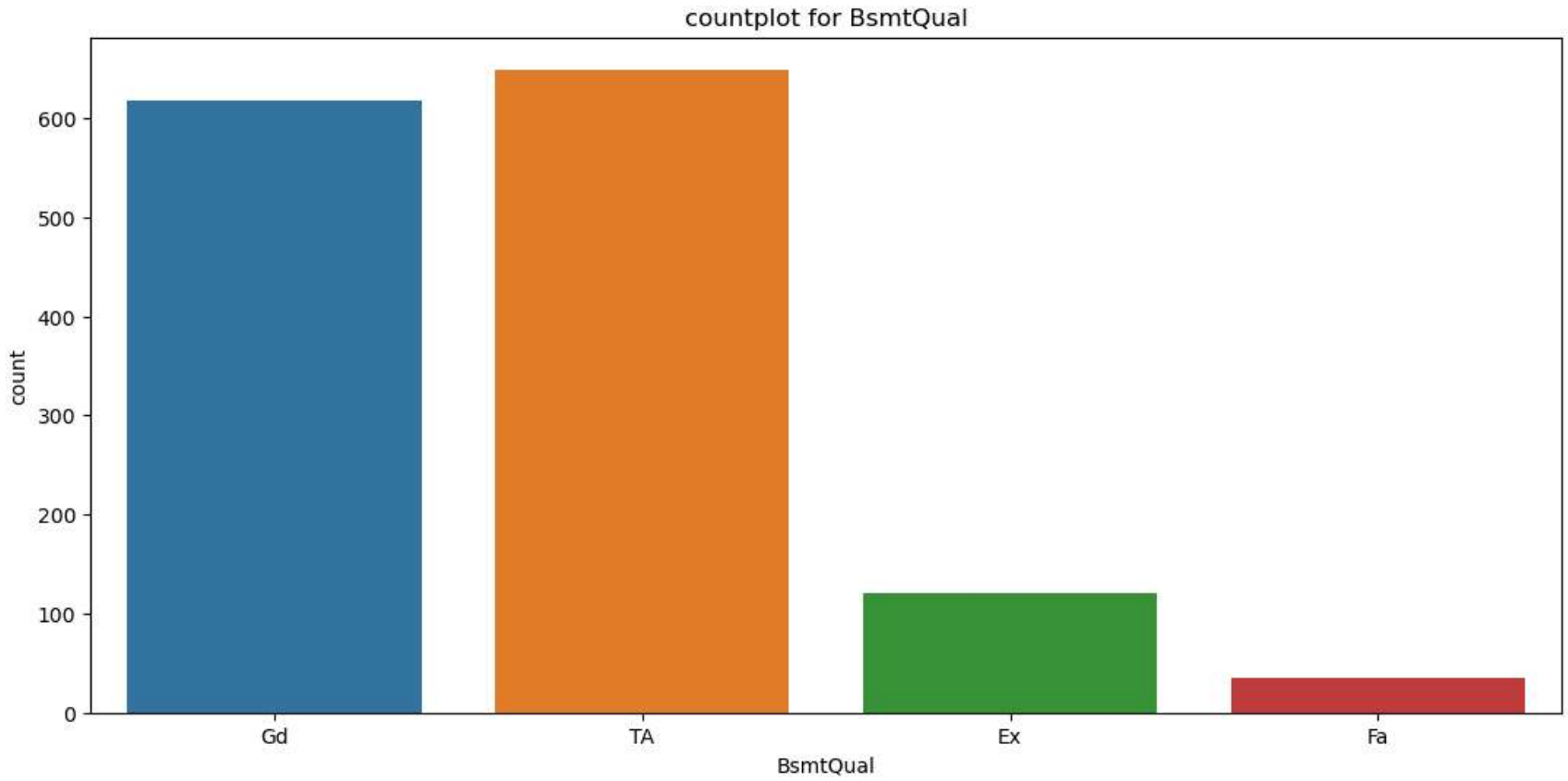

countplot for Exterior2nd
count
0
100
200
300
400
500
VinylSd
MetalSd
Wd Shng
HdBoard
Plywood
Wd Sdng
CmentBd
BrkFace
Stucco
AsbShng
Brk Cmn
ImStucc
AsphShn
Stone
Other
CBlock
Exterior2nd

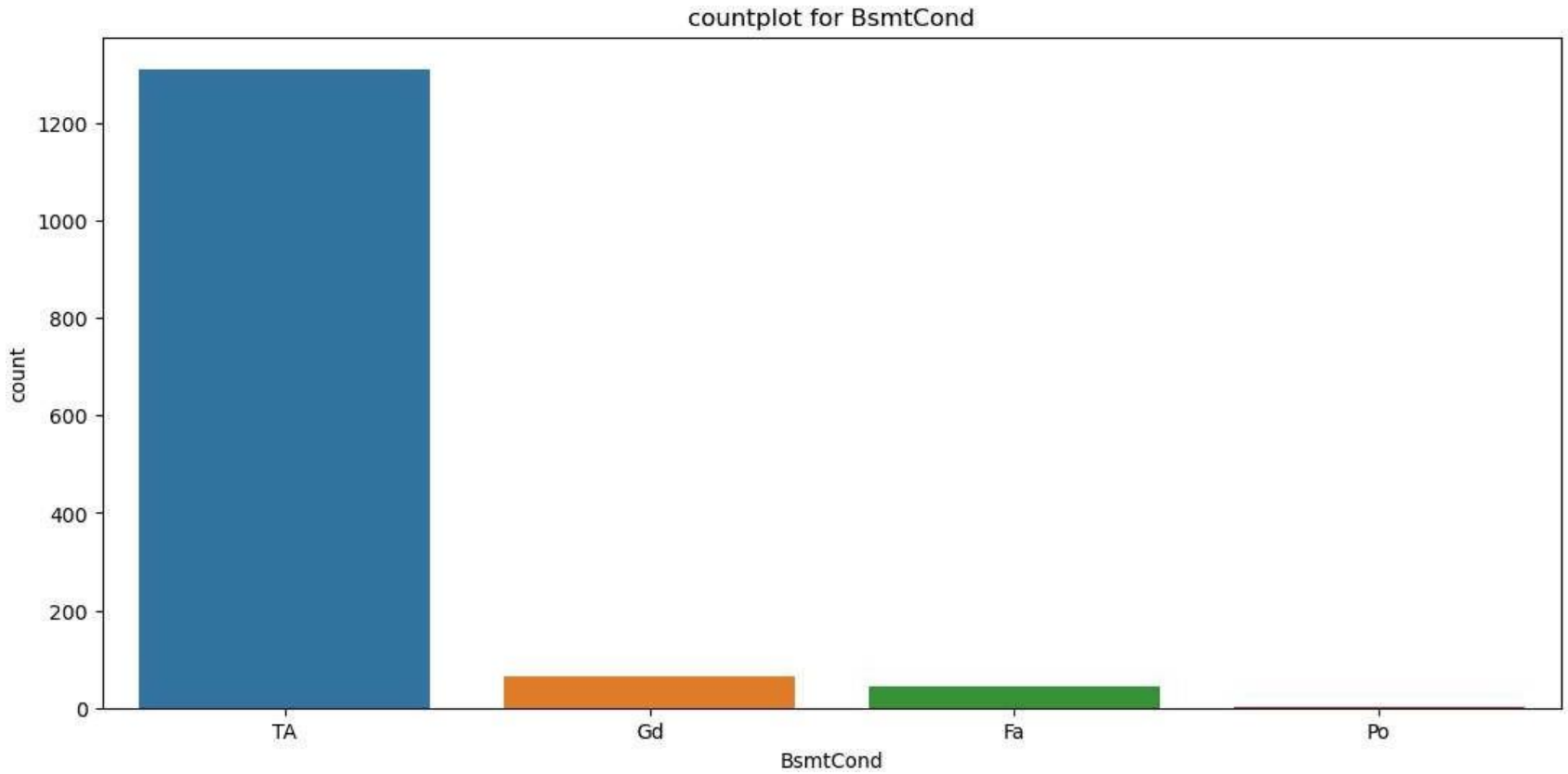

countplot for MasVnrType
count
400
300
200
100
0
BrkFace
Stone
BrkCmn
MasVnrType

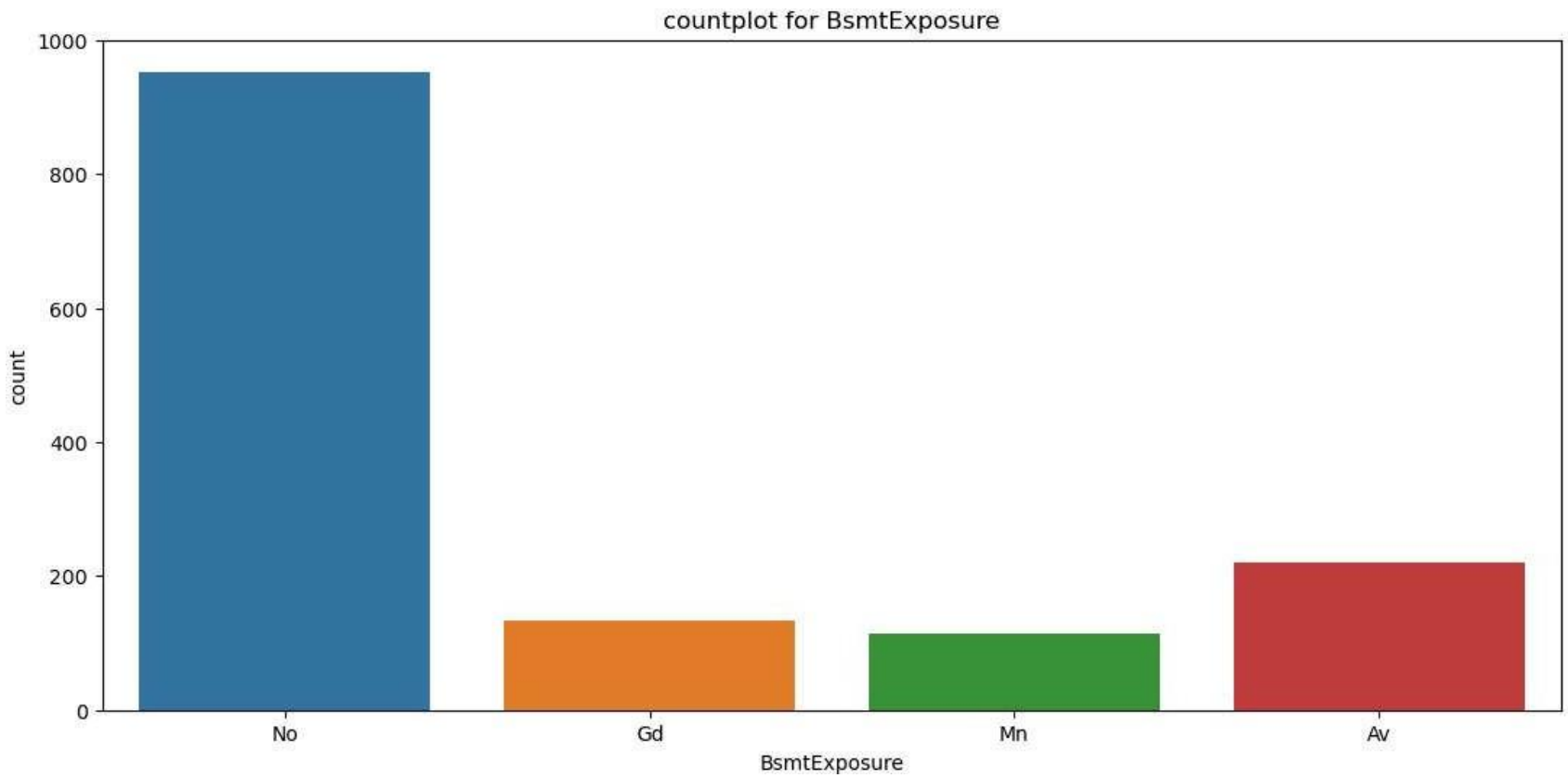

countplot for ExterQual
count
800
600
400
200
0
Gd
TA
Ex
Fa
ExterQual

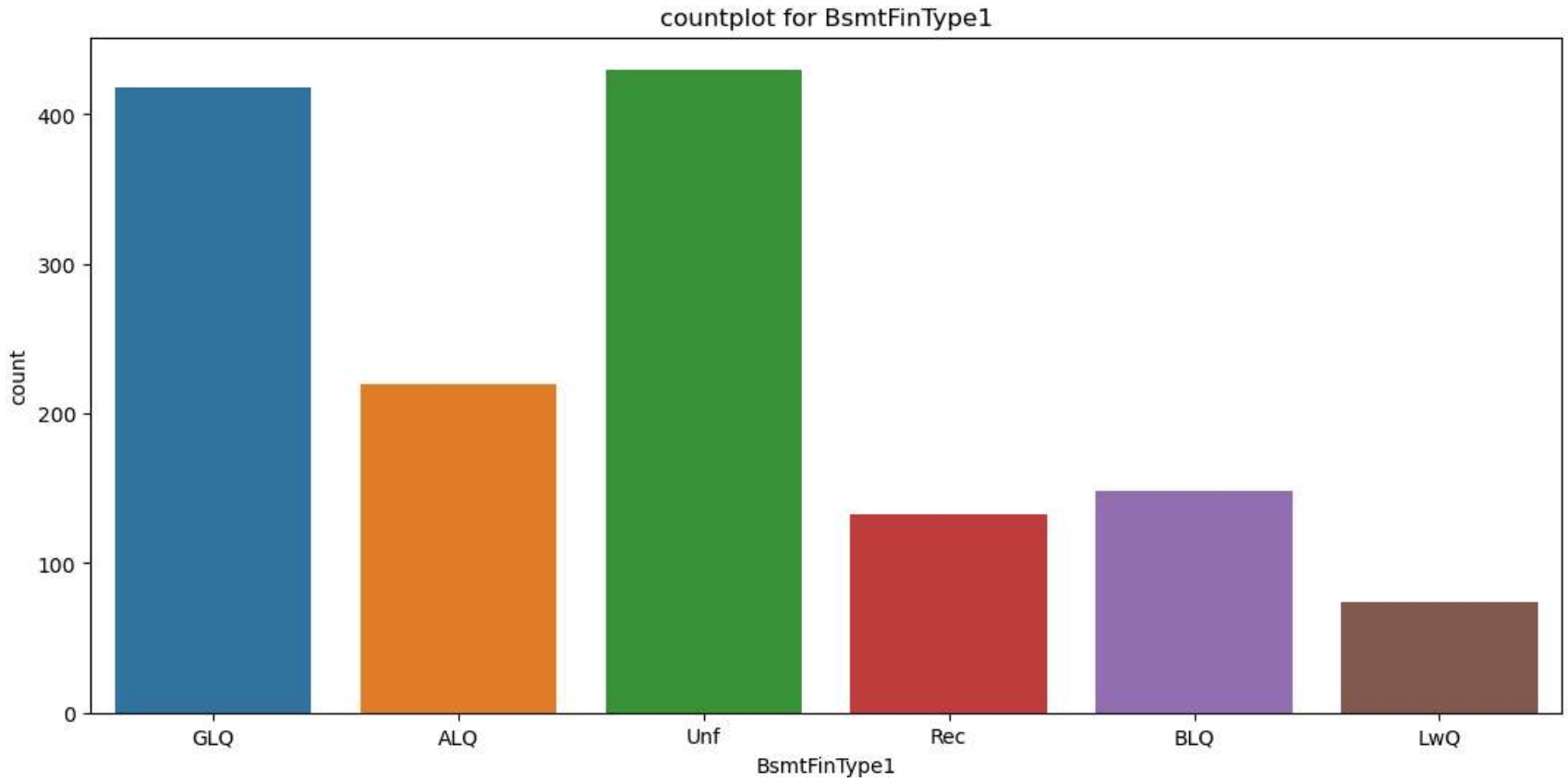

countplot for ExterCond
count
1200
1000
800
600
400
200
0
TA
Gd
Fa
Po
Ex
ExterCond

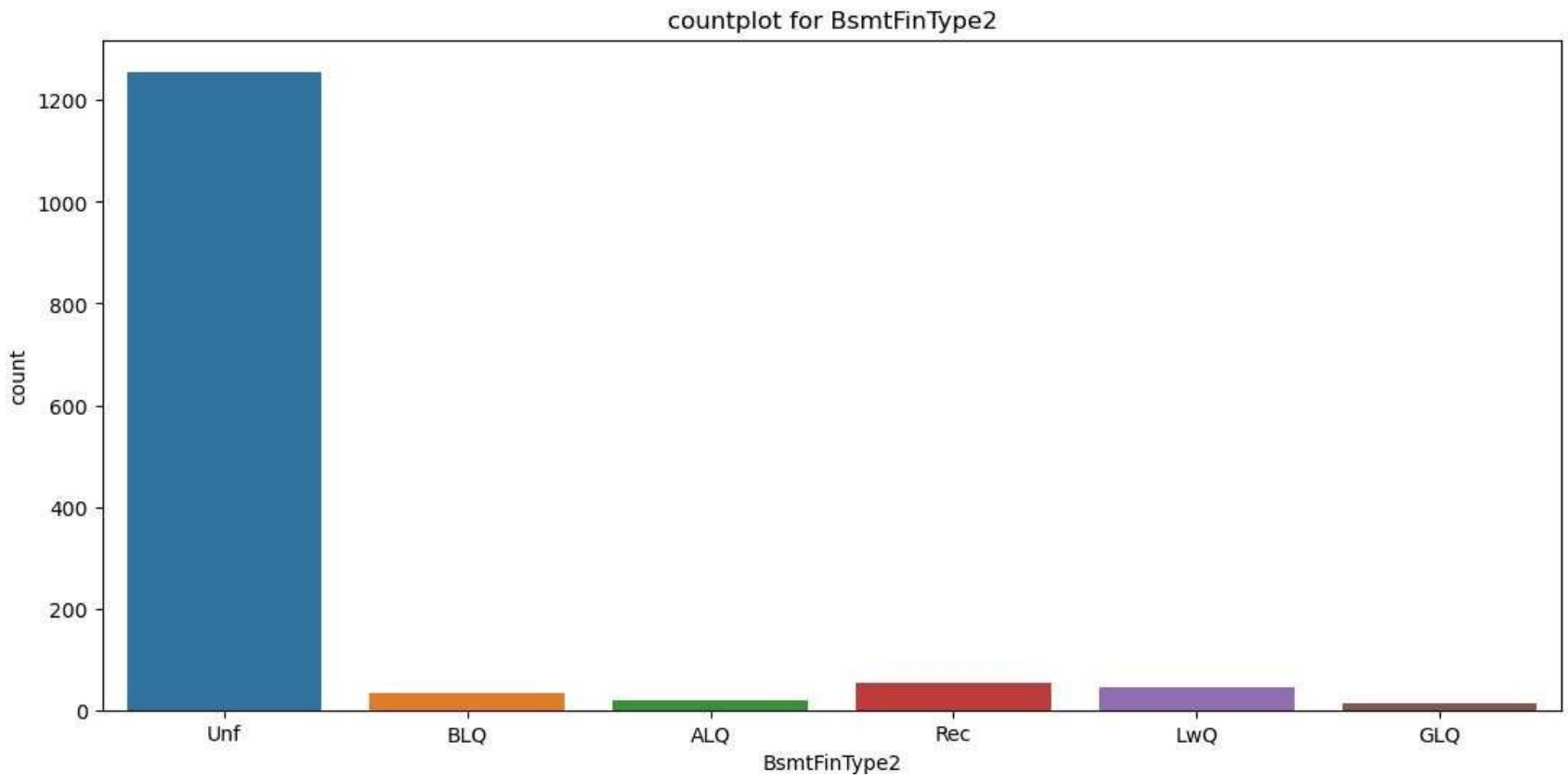

countplot for Foundation
count
0
100
200
300
400
500
600
PConc
CBlock
BrkTil
Wood
Slab
Stone
Foundation

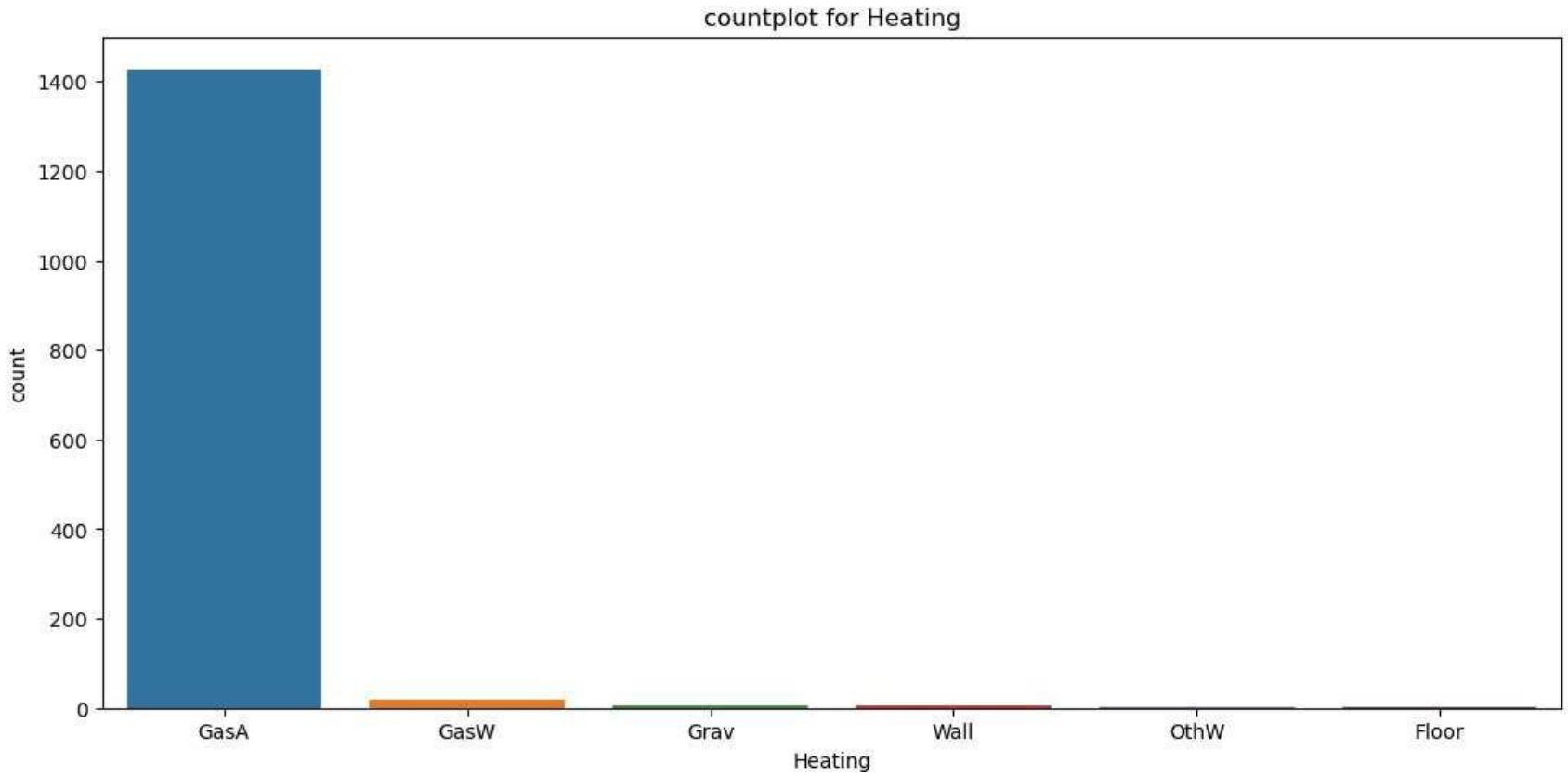

countplot for BsmtQual
count
0
100
200
300
400
500
600
Gd
TA
Ex
Fa
BsmtQual

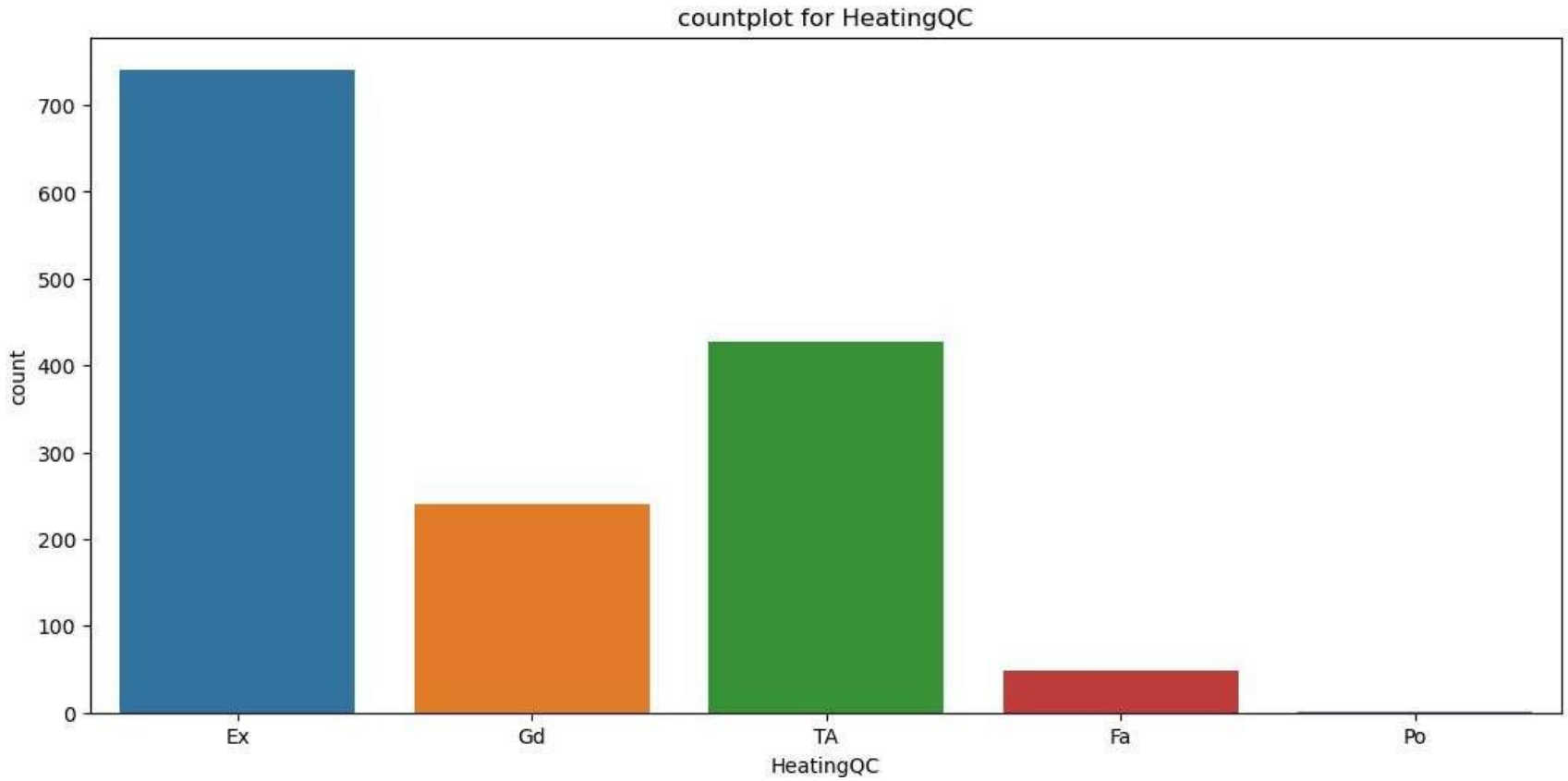

countplot for BsmtCond
count
1200
1000
800
600
400
200
0
TA
Gd
Fa
Po
BsmtCond

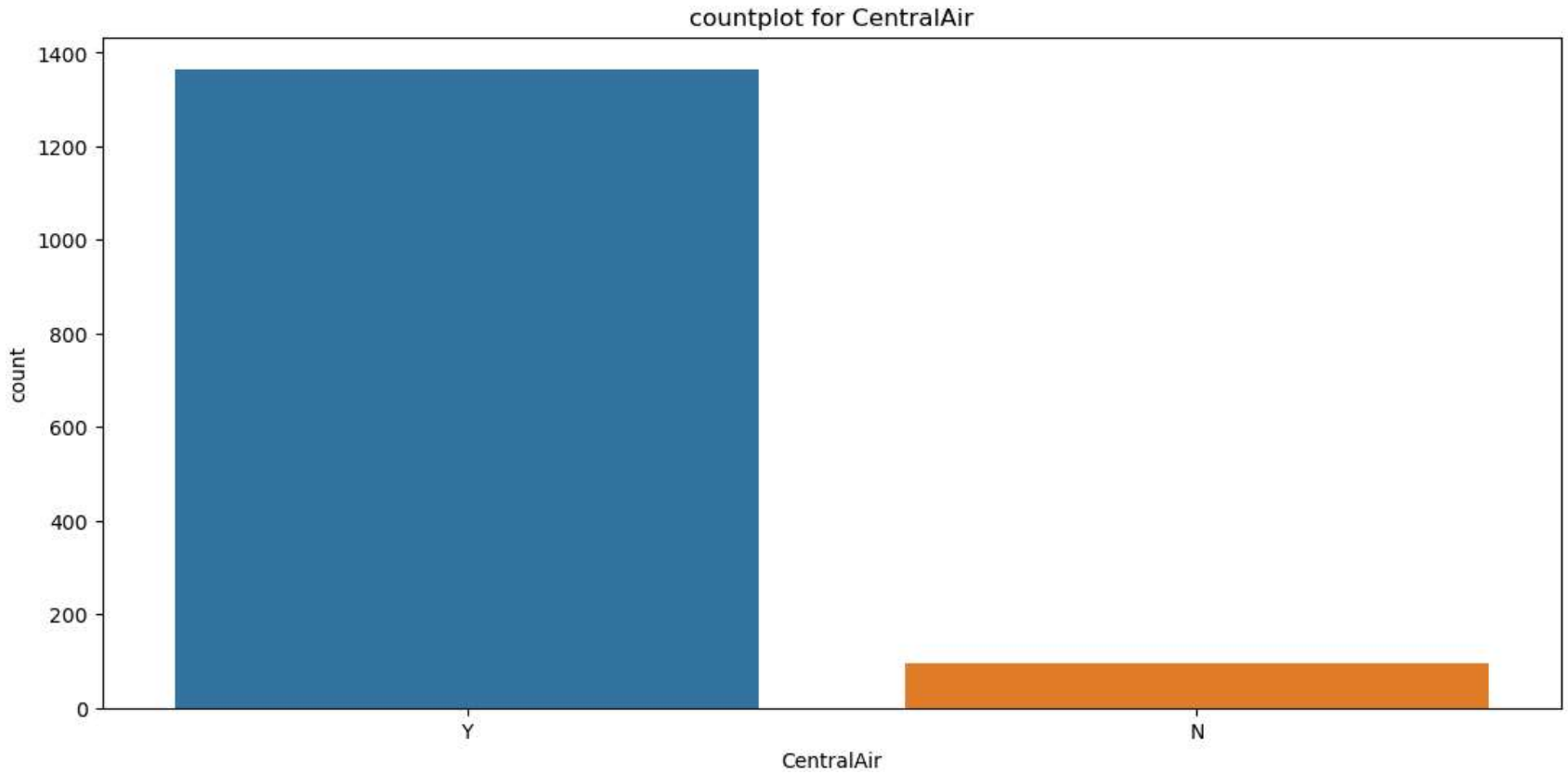

countplot for BsmtExposure
1000
800
600
400
200
0
count
No
Gd
Mn
Av
BsmtExposure

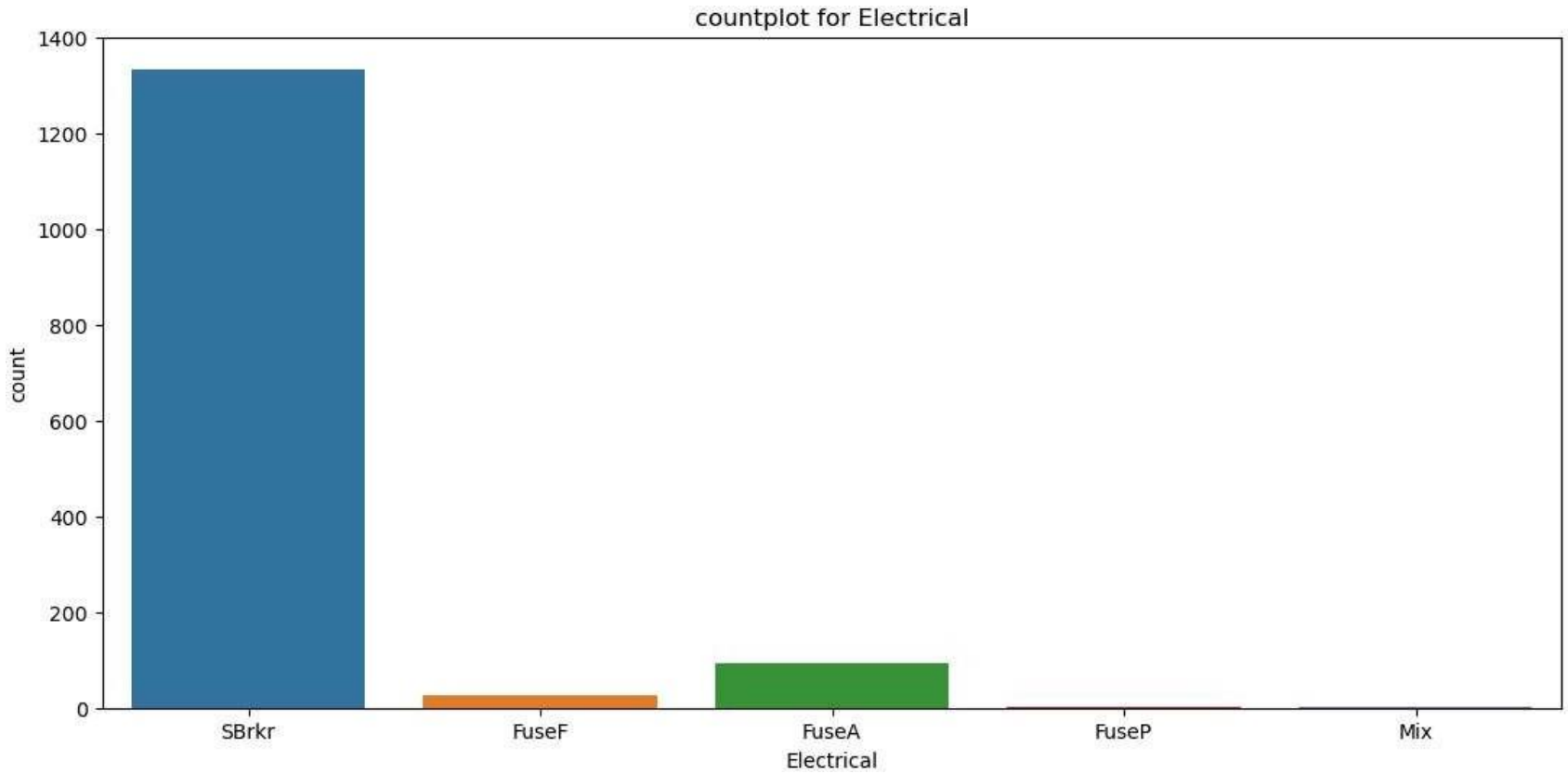

countplot for BsmtFinType1
count
0
100
200
300
400
GLQ
ALQ
Unf
Rec
BLQ
LwQ
BsmtFinType1

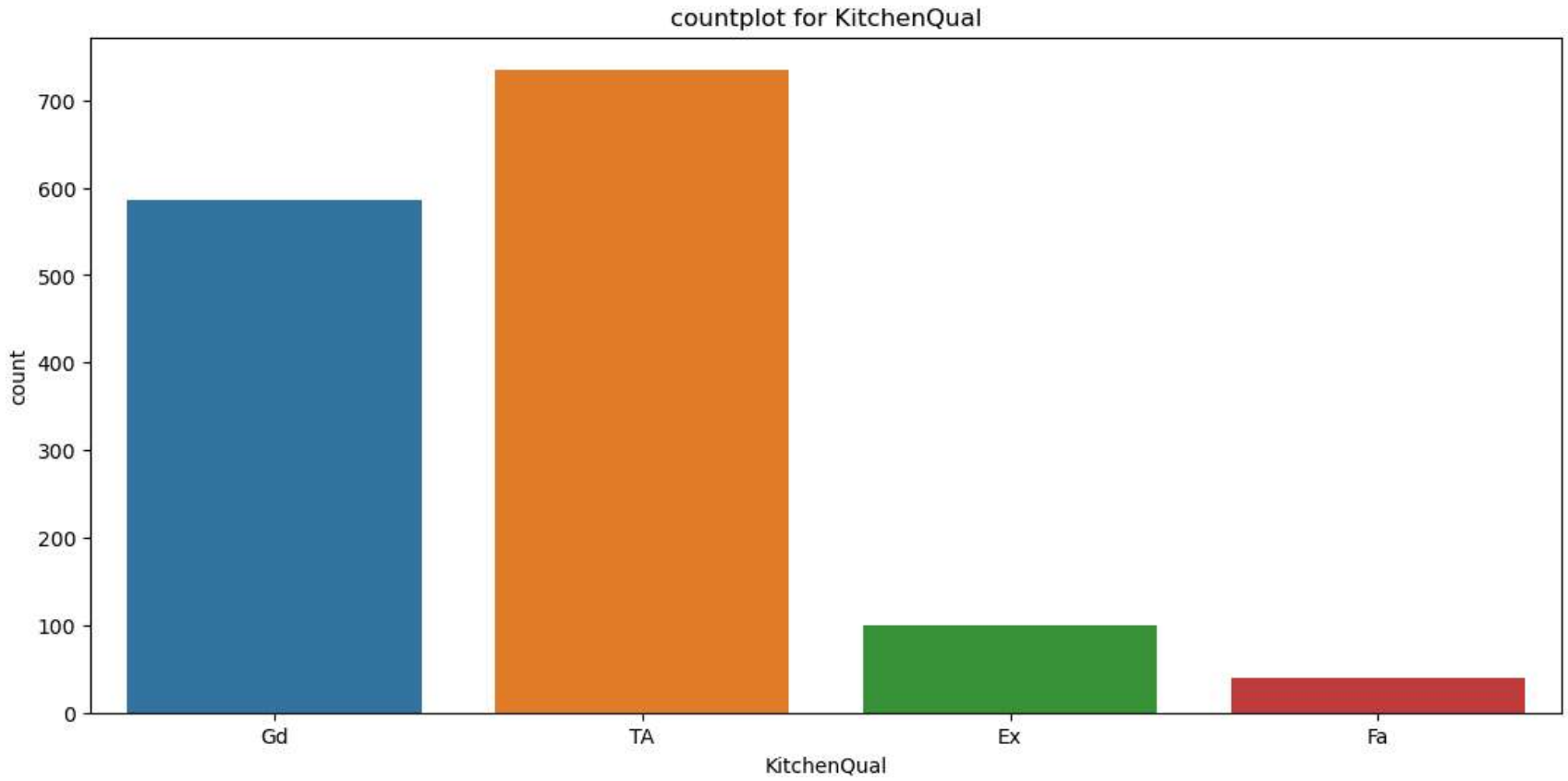

countplot for BsmtFinType2
count
0
200
400
600
800
1000
1200
Unf
BLQ
ALQ
Rec
LwQ
GLQ
BsmtFinType2

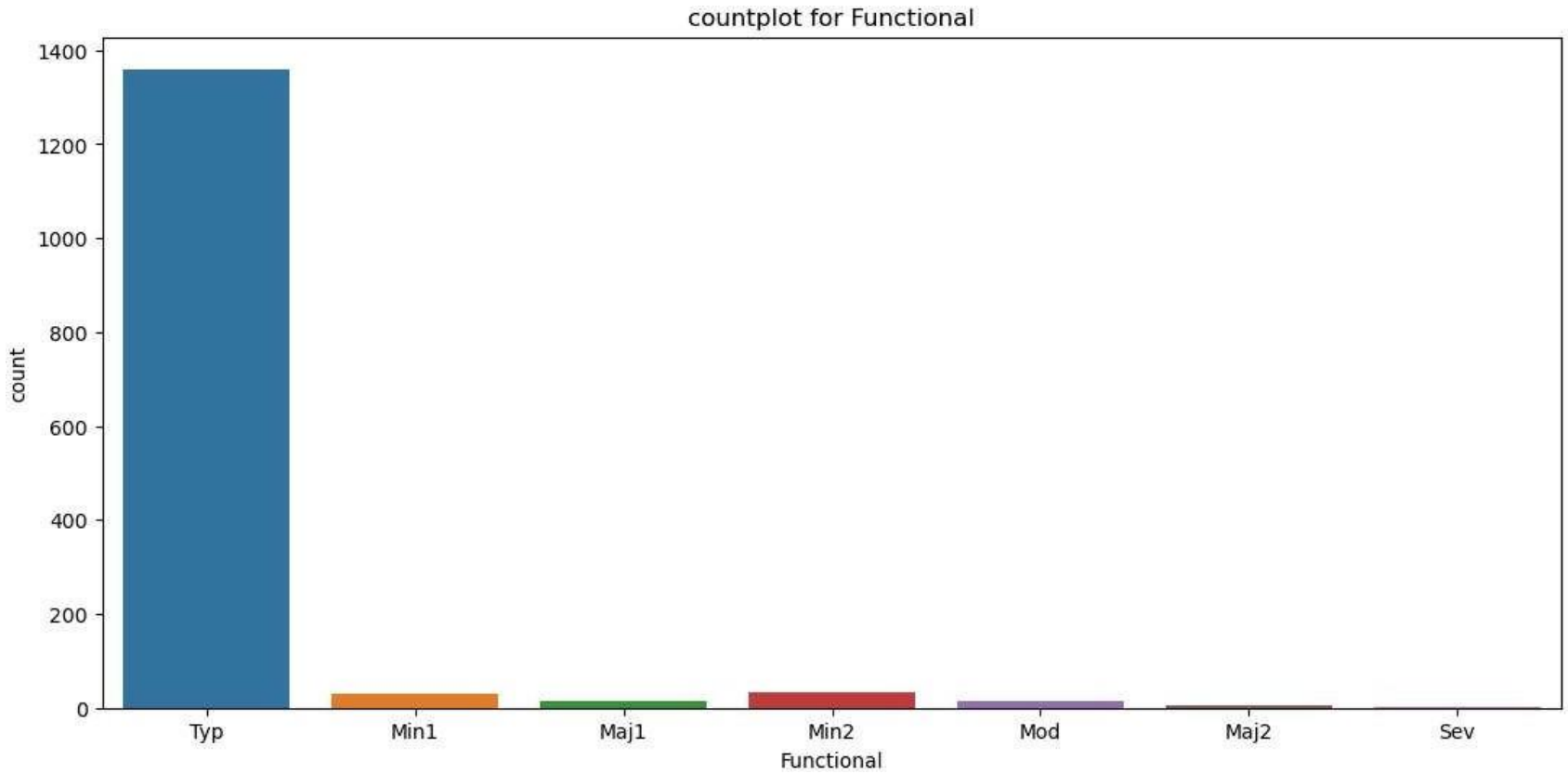

countplot for Heating
count
1400
1200
1000
800
600
400
200
0
GasA
GasW
Grav
Wall
OthW
Floor
Heating

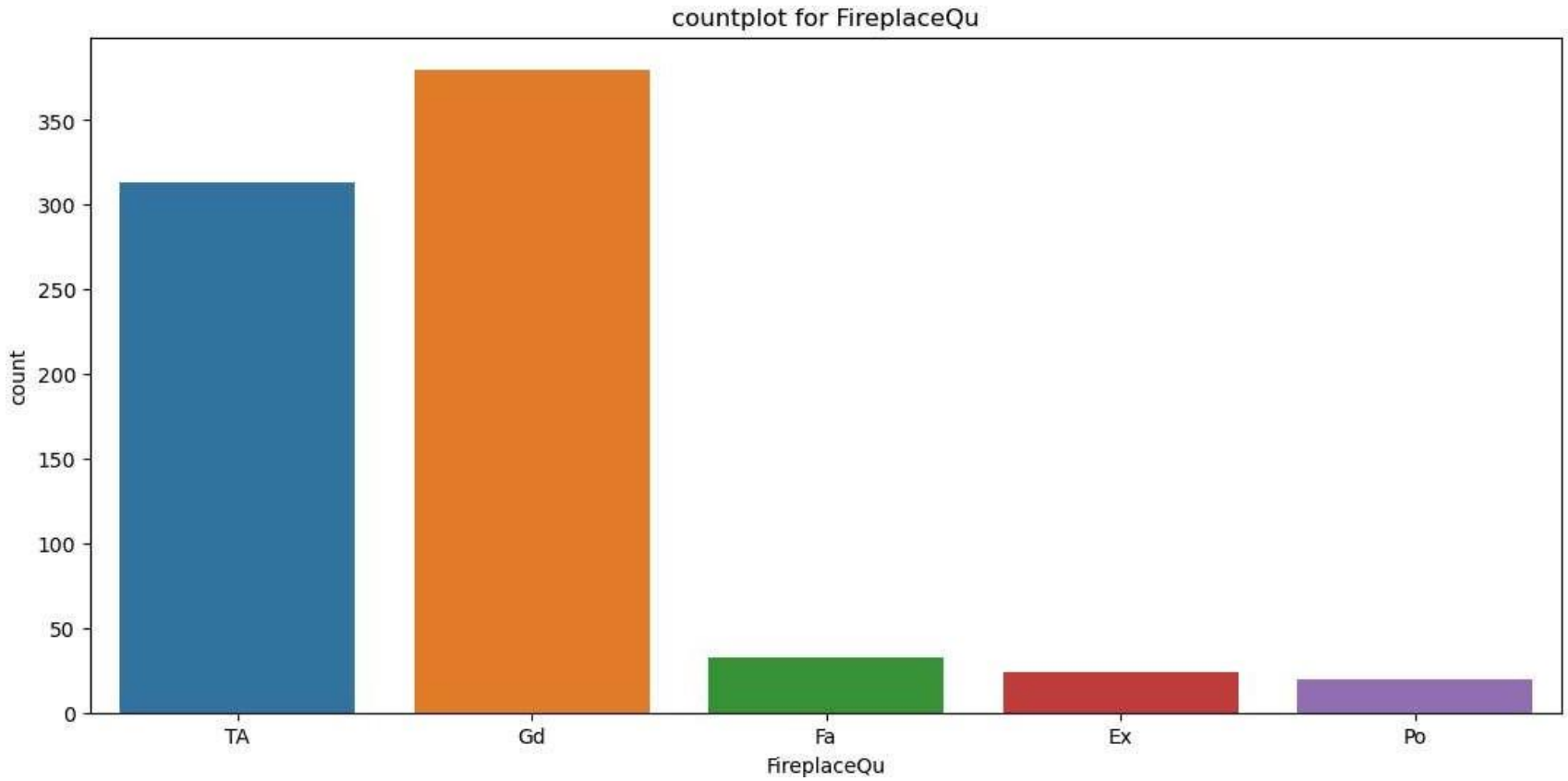

countplot for HeatingQC
count
0
100
200
300
400
500
600
700
Ex
Gd
TA
Fa
Po
HeatingQC

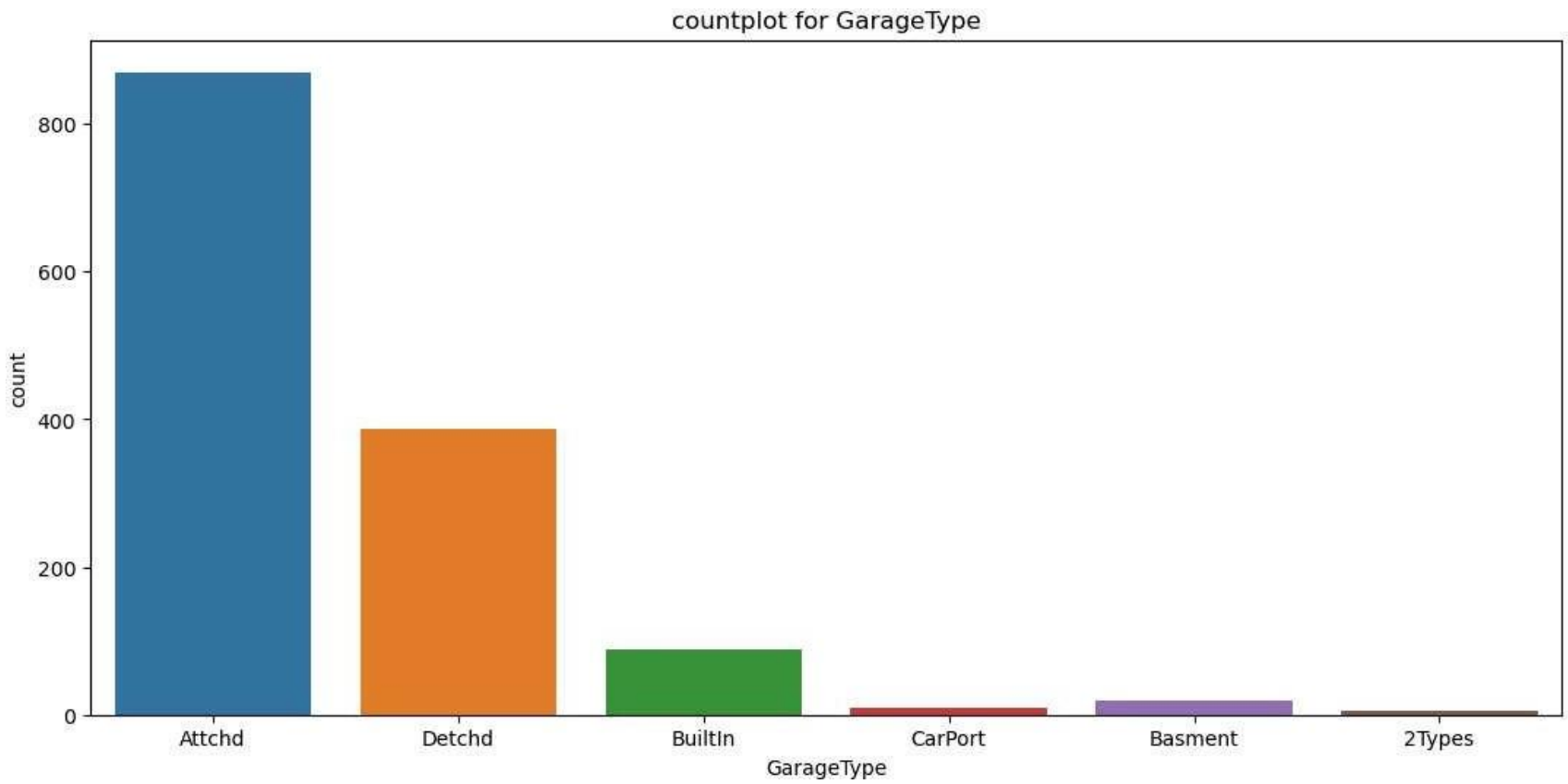

countplot for CentralAir
1400
1200
1000
800
600
400
200
0
count
Y
N
CentralAir

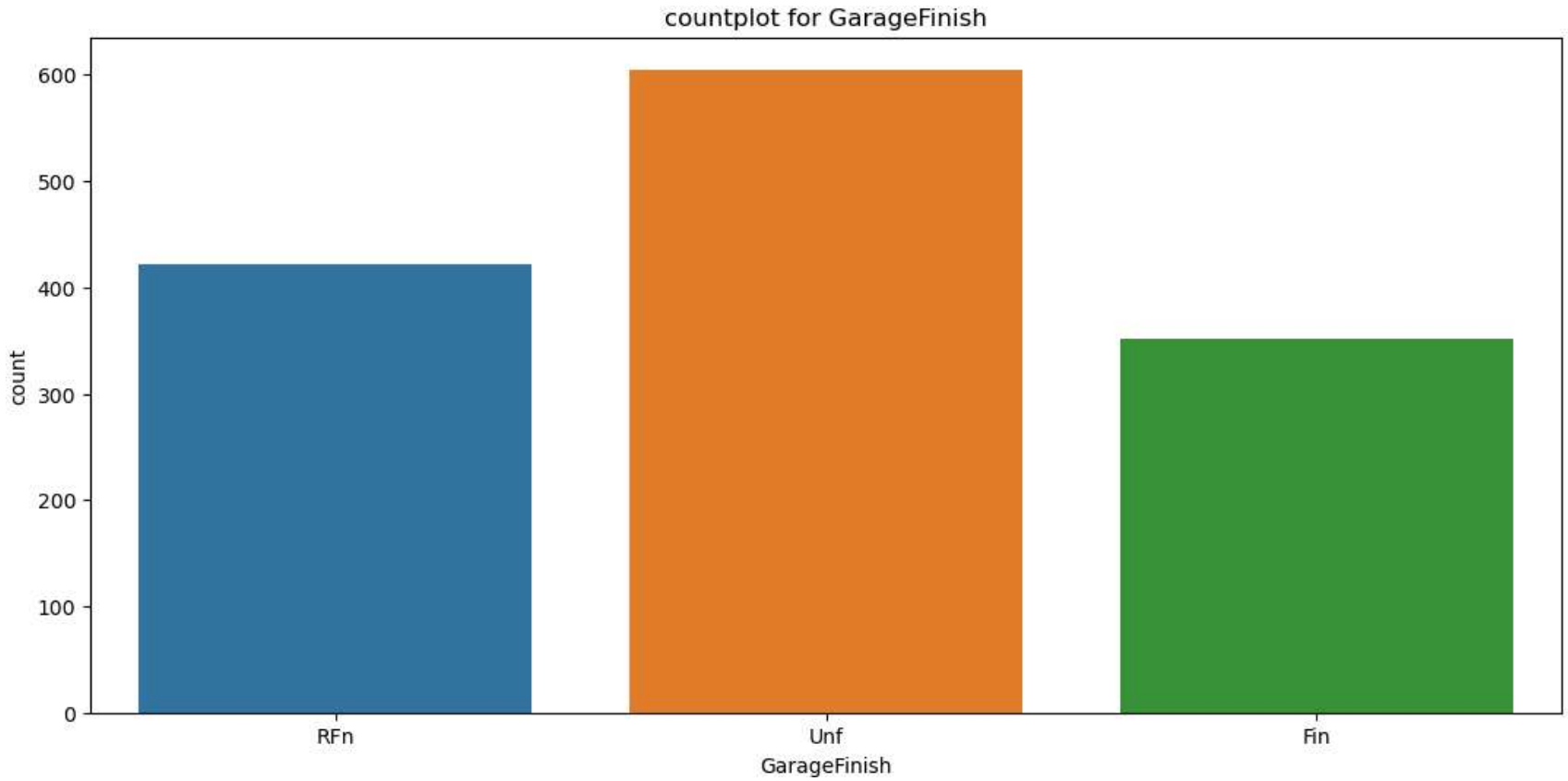

countplot for Electrical
1400
1200
1000
800
600
400
200
0
count
SBrkr
FuseF
FuseA
FuseP
Mix
Electrical

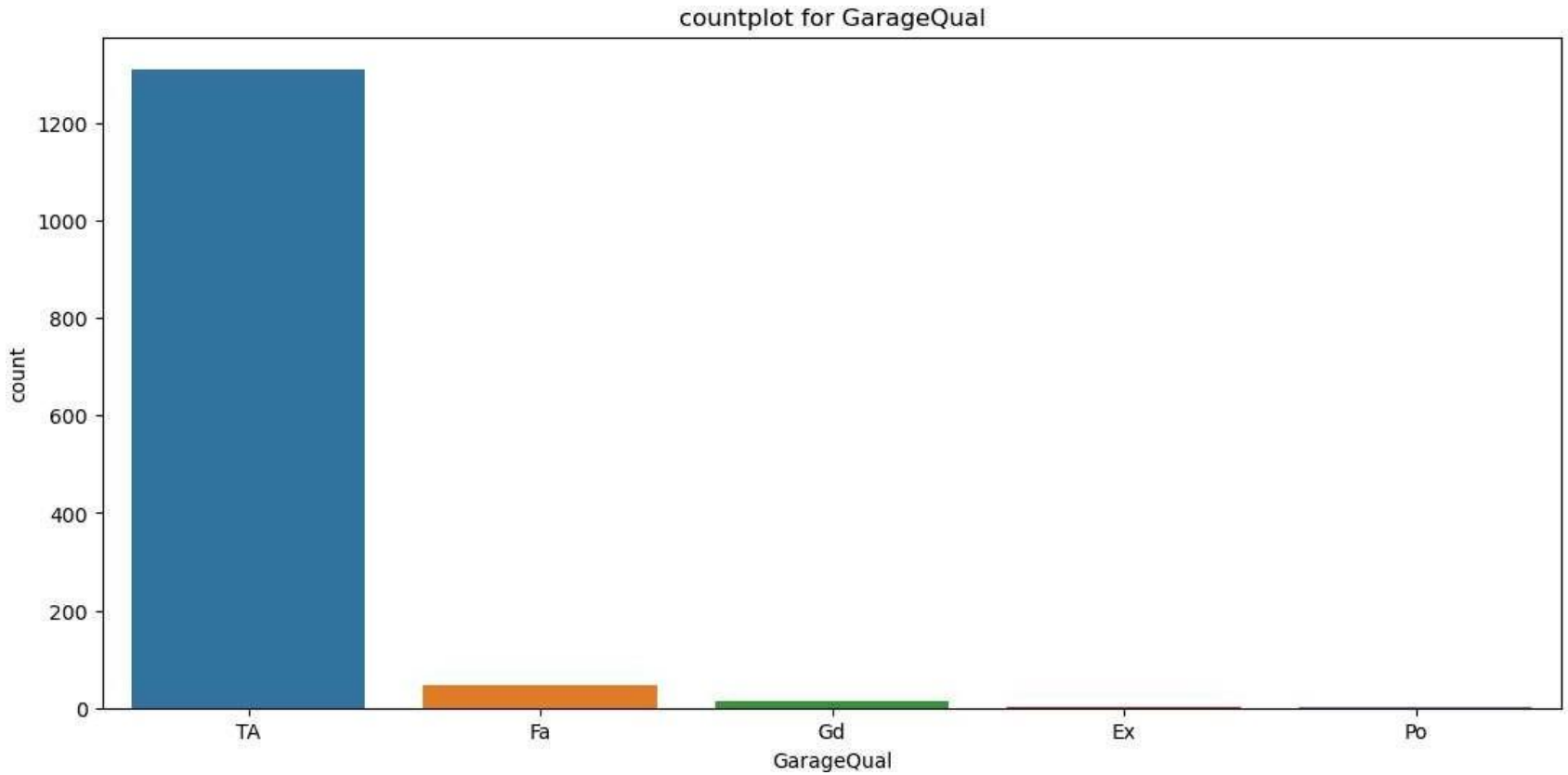

countplot for KitchenQual
count
0
100
200
300
400
500
600
700
Gd
TA
Ex
Fa
KitchenQual

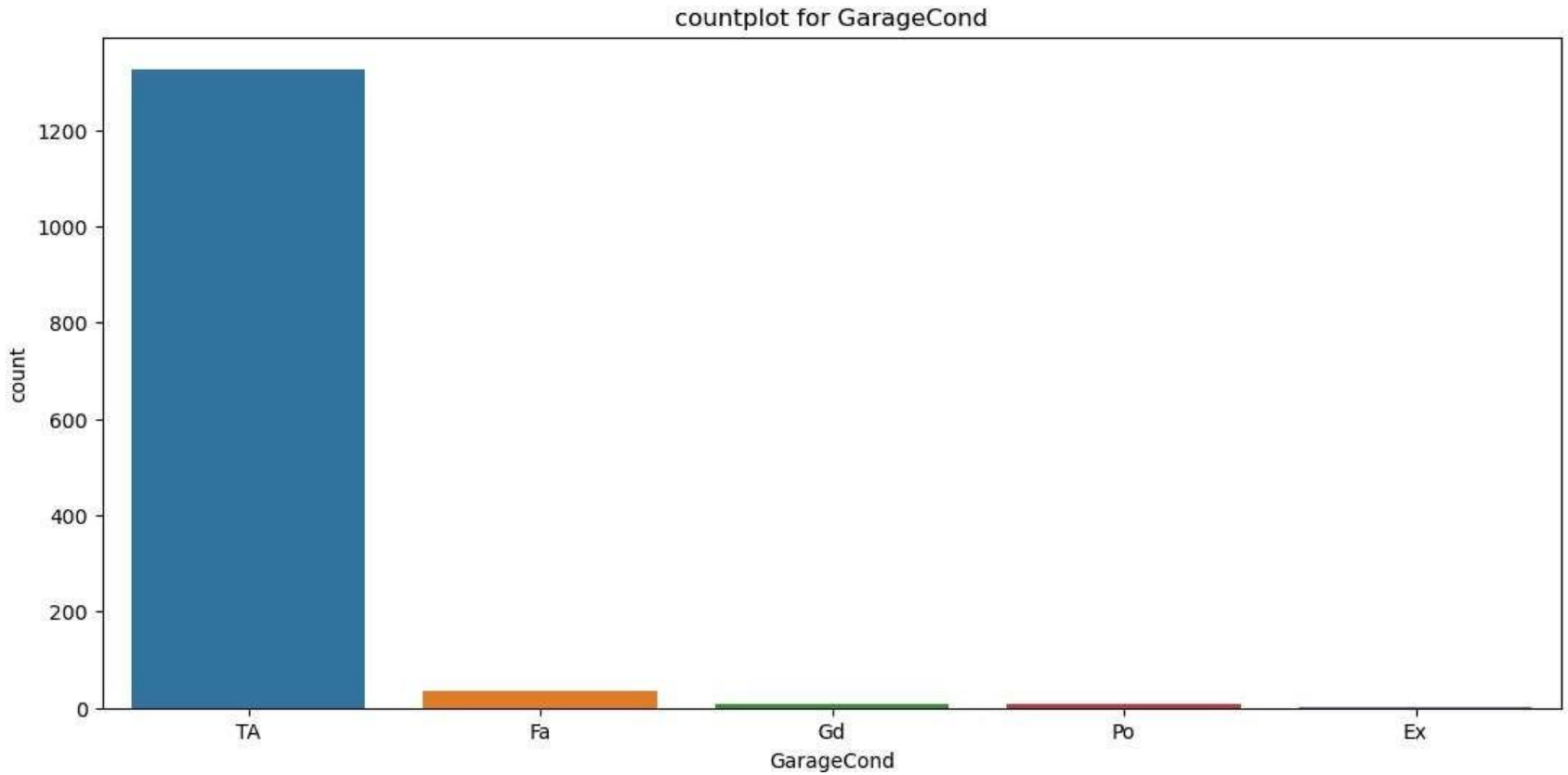

countplot for Functional
1400
1200
1000
800
600
400
200
0
count
Typ
Min1
Maj1
Min2
Mod
Maj2
Sev
Functional

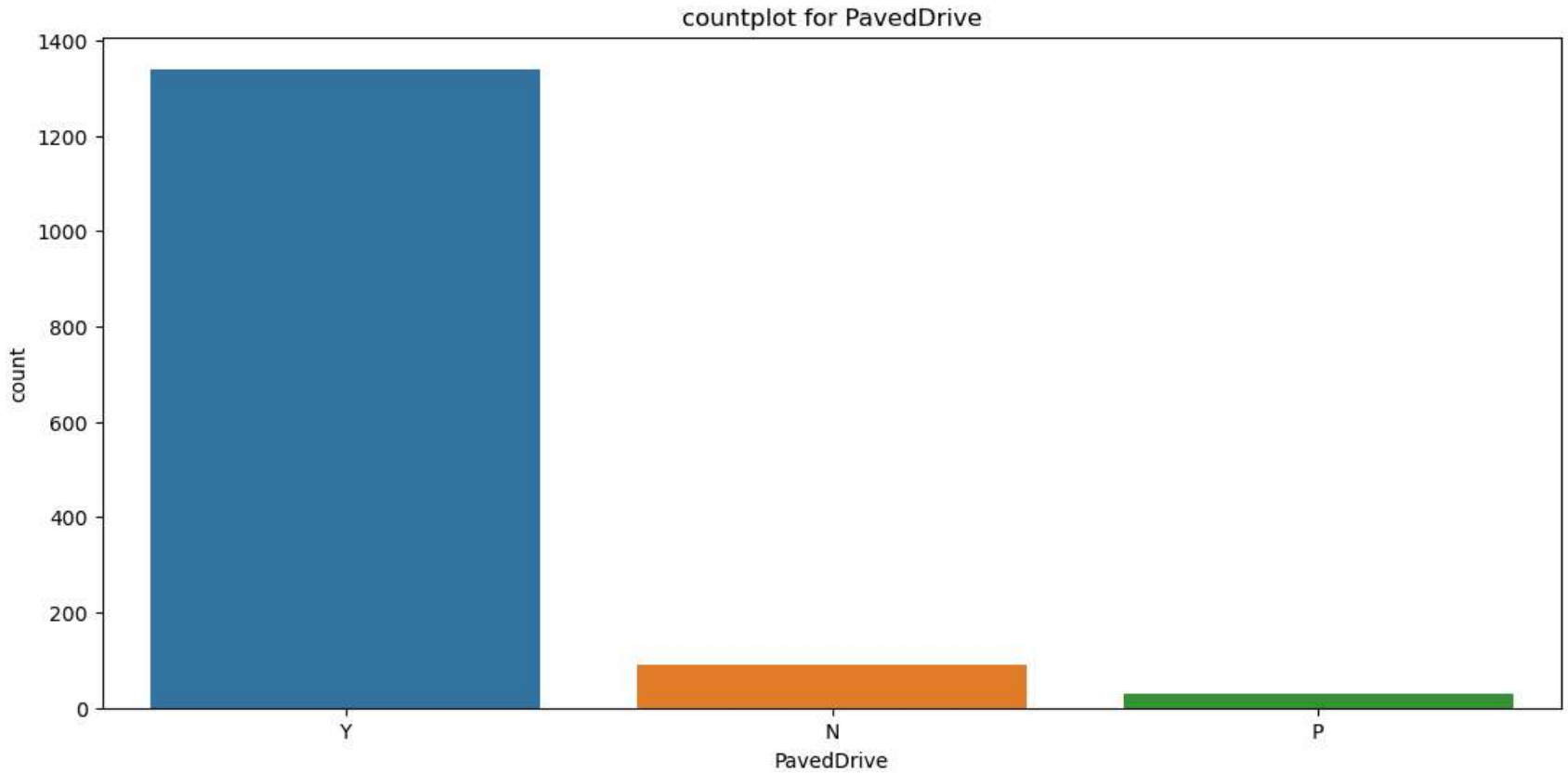

countplot for FireplaceQu
count
0
50
100
150
200
250
300
350
TA
Gd
Fa
Ex
Po
FireplaceQu

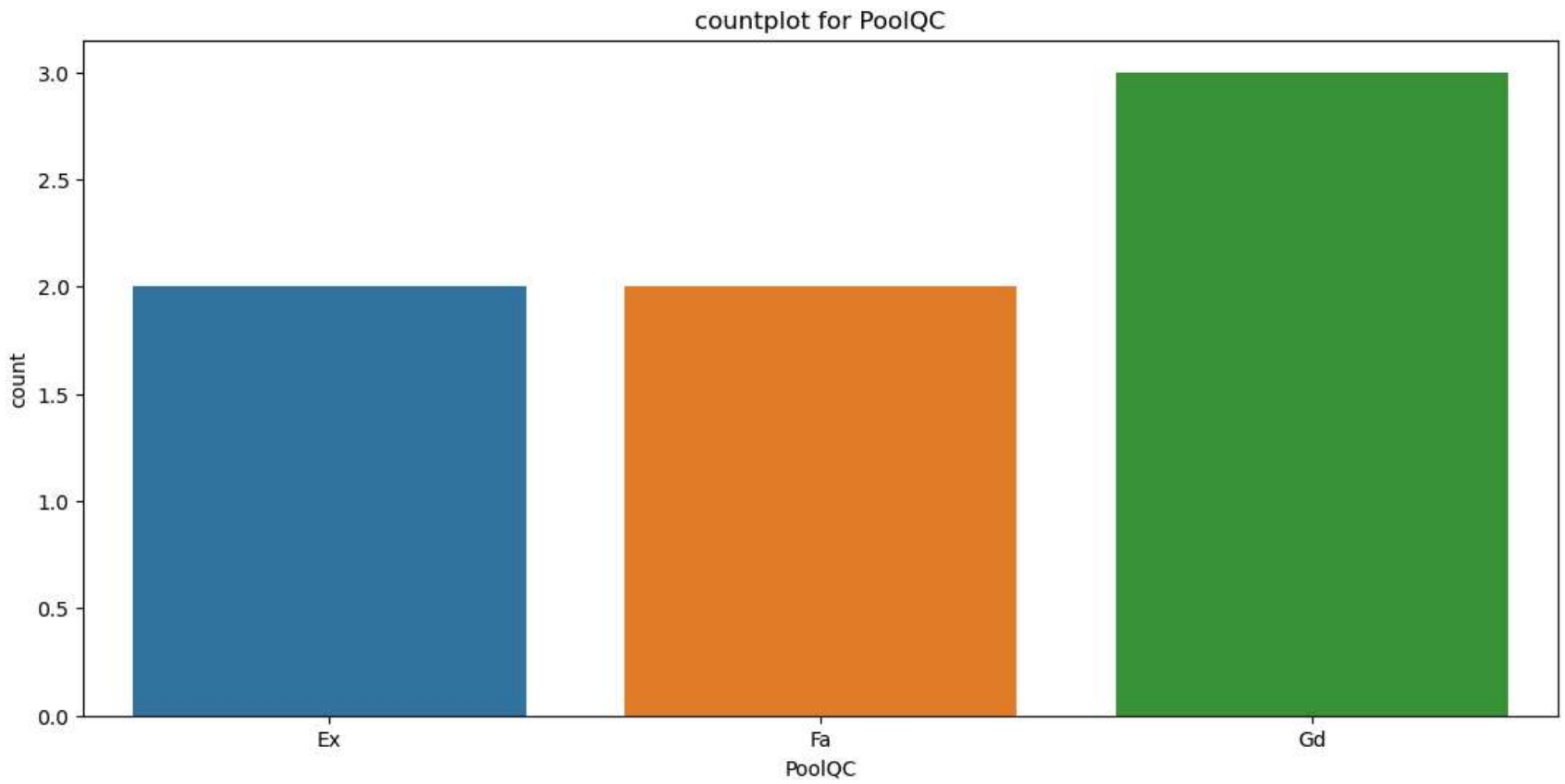

countplot for GarageType
count
800
600
400
200
0
Attchd
Detchd
BuiltIn
CarPort
Basment
2Types
GarageType

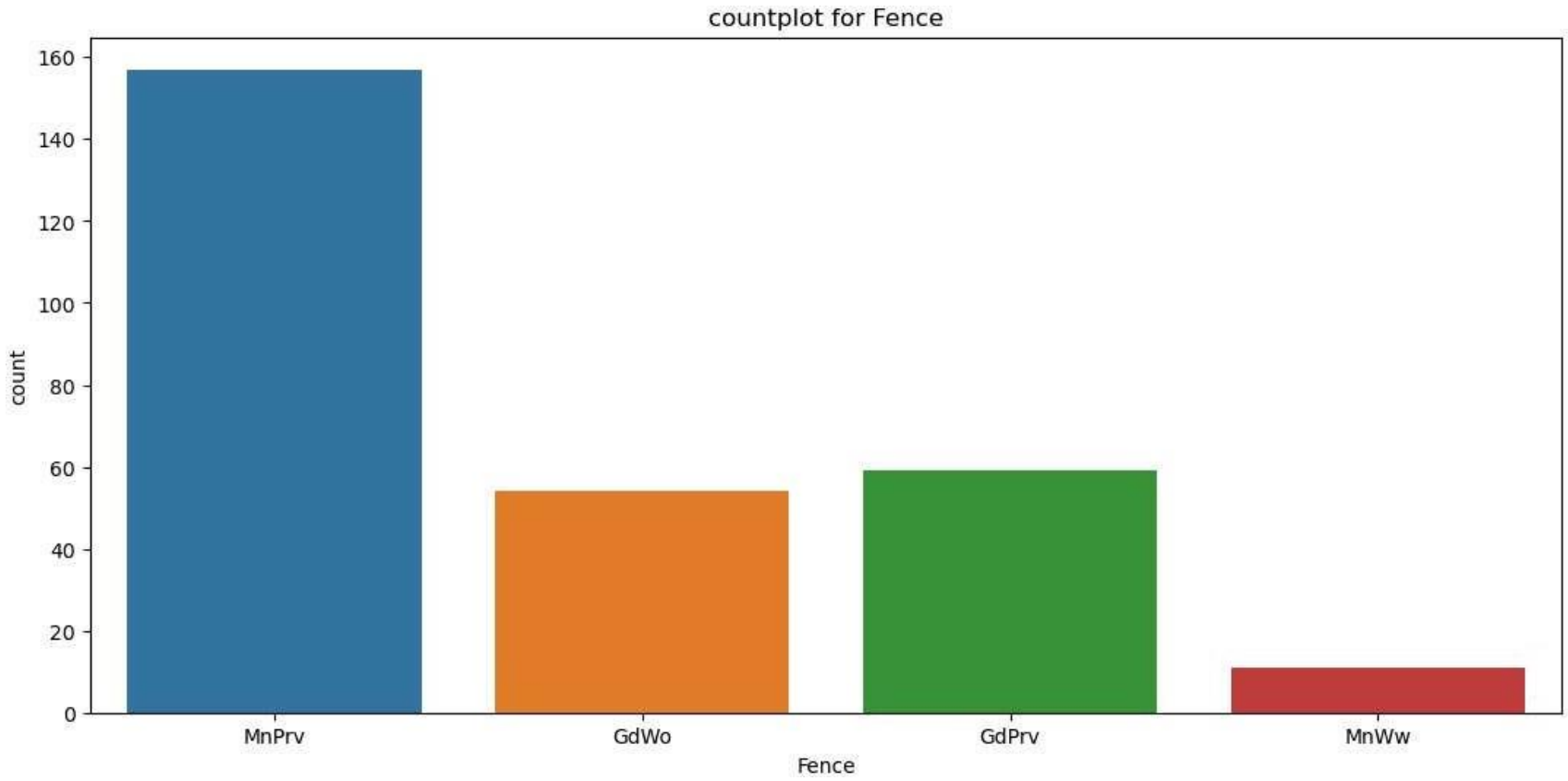

countplot for GarageFinish
600
500
400
300
200
100
0
count
RFn
Unf
Fin
GarageFinish

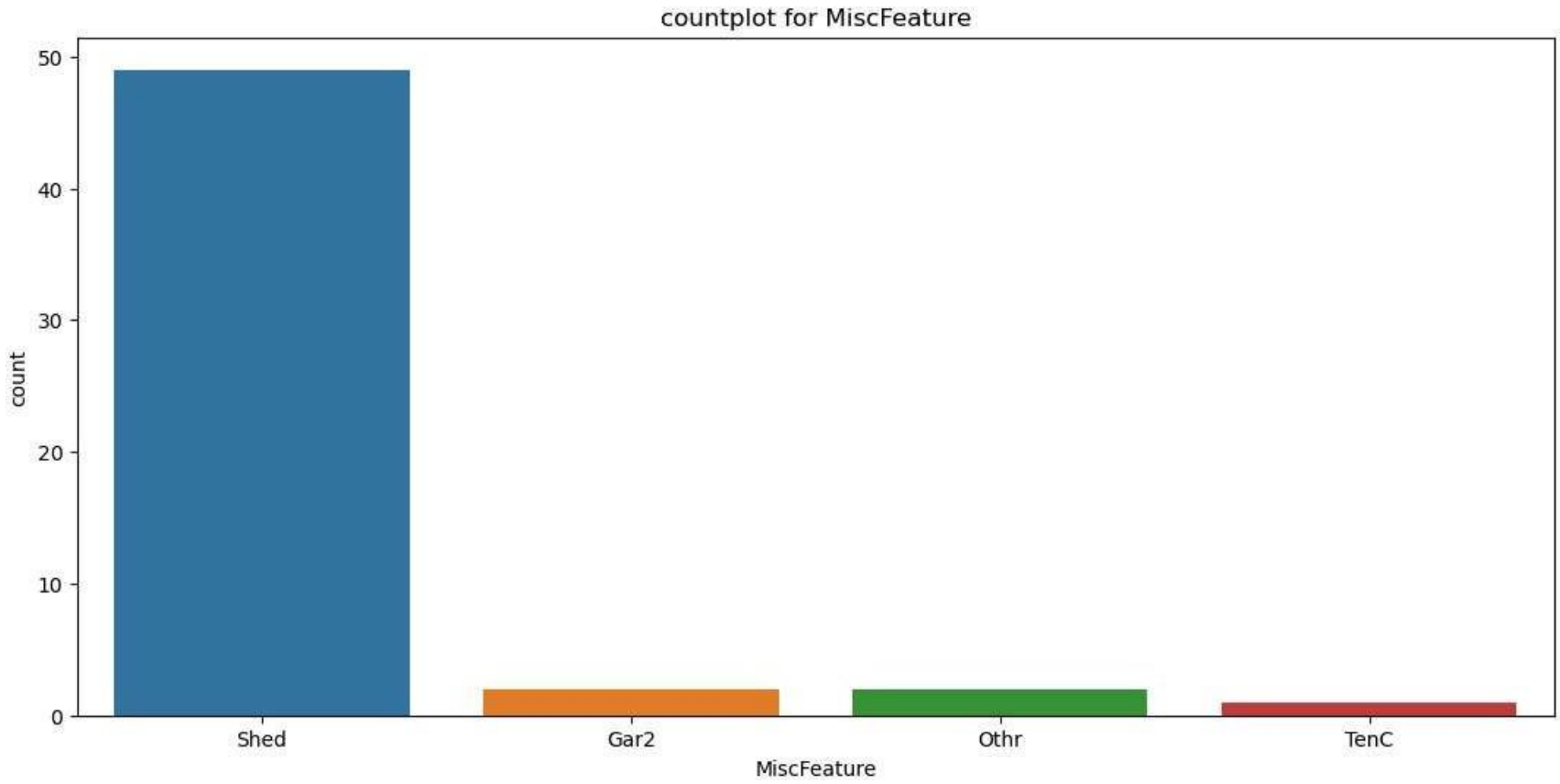

countplot for GarageQual
count
1200
1000
800
600
400
200
0
TA
Fa
Gd
Ex
Po
GarageQual

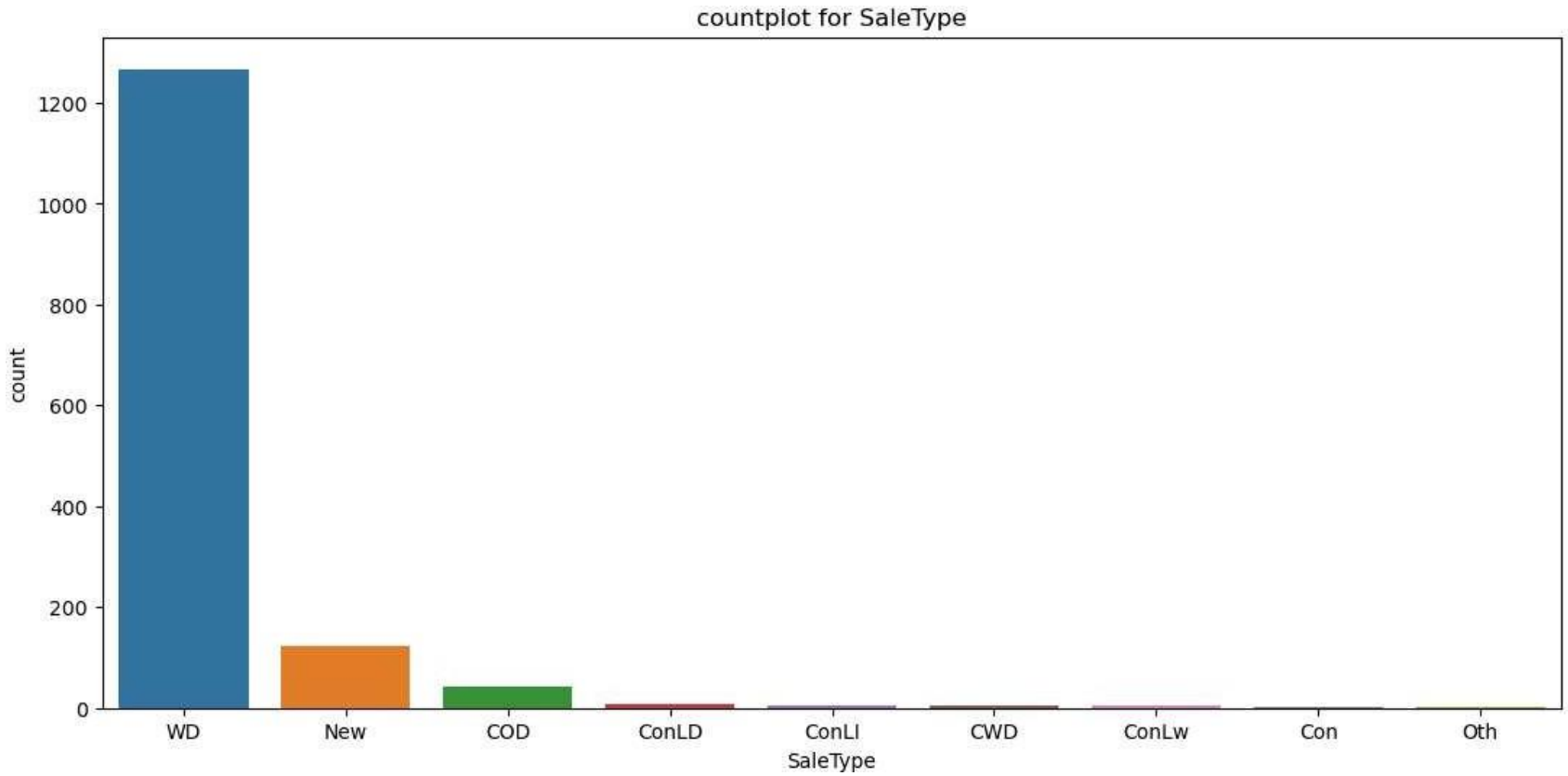

countplot for GarageCond
count
1200
1000
800
600
400
200
0
TA
Fa
Gd
Po
Ex
GarageCond

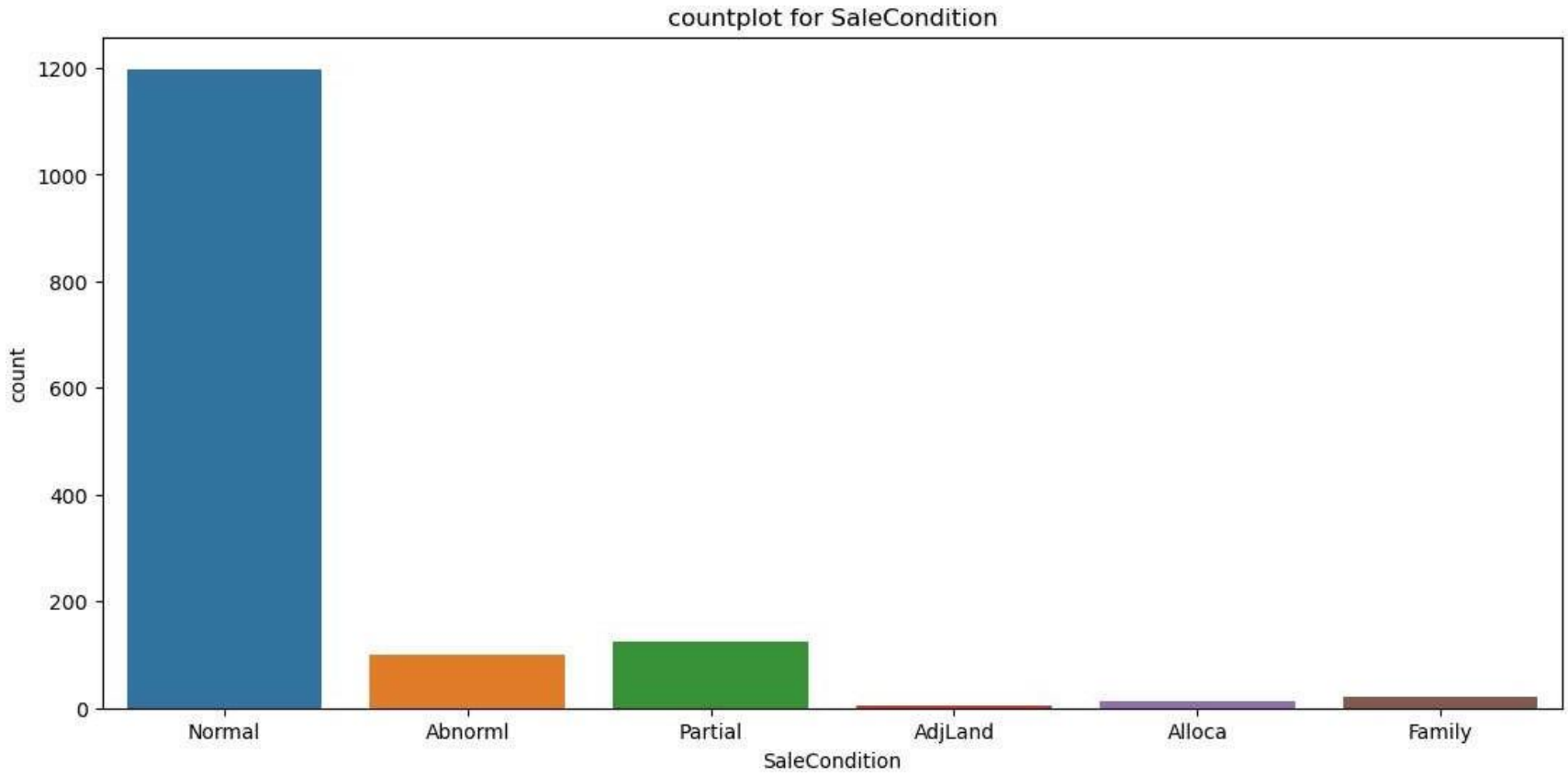

countplot for PavedDrive
1400
1200
1000
800
600
400
200
0
count
Y
N
P
PavedDrive


countplot for PoolQC
count
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Ex
Fa
Gd
PoolQC


countplot for Fence
count
Fence
MnPrv
GdWo
GdPrv
MnWw
0
20
40
60
80
100
120
140
160

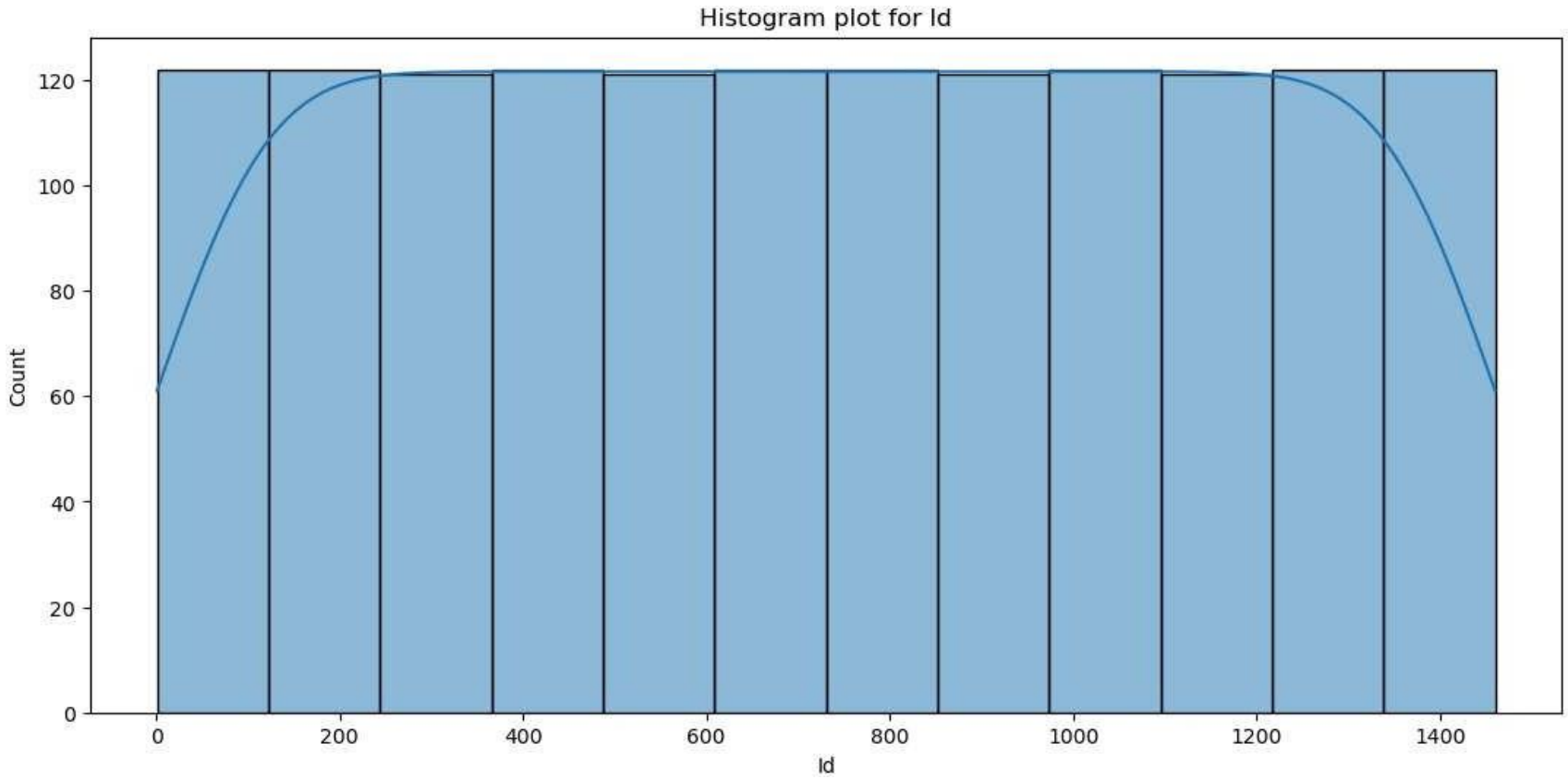

countplot for MiscFeature
count
50
40
30
20
10
0
Shed
Gar2
Othr
TenC
MiscFeature

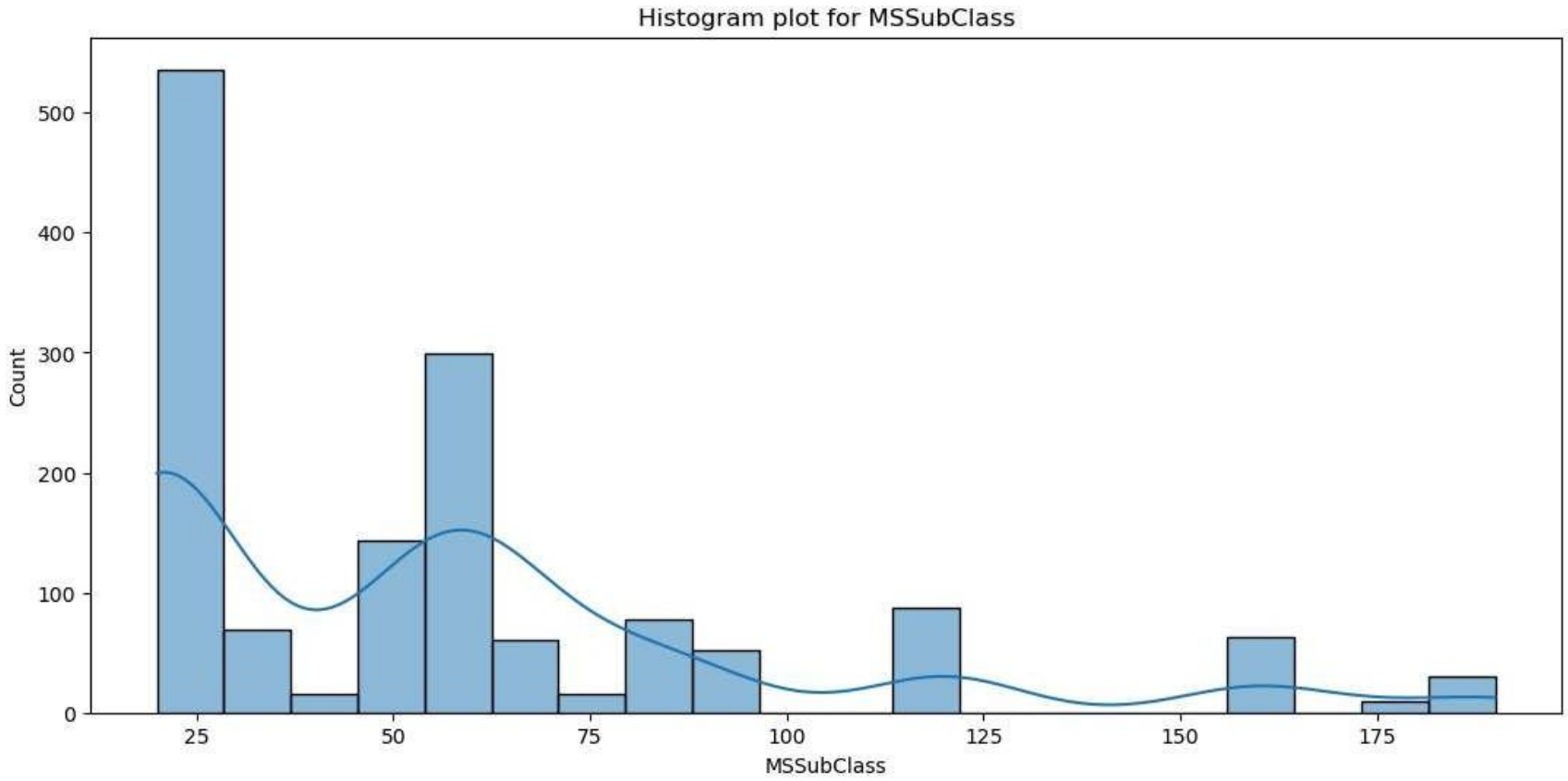

countplot for SaleType
count
1200
1000
800
600
400
200
0
WD
New
COD
ConLD
ConLI
CWD
ConLw
Con
Oth
SaleType

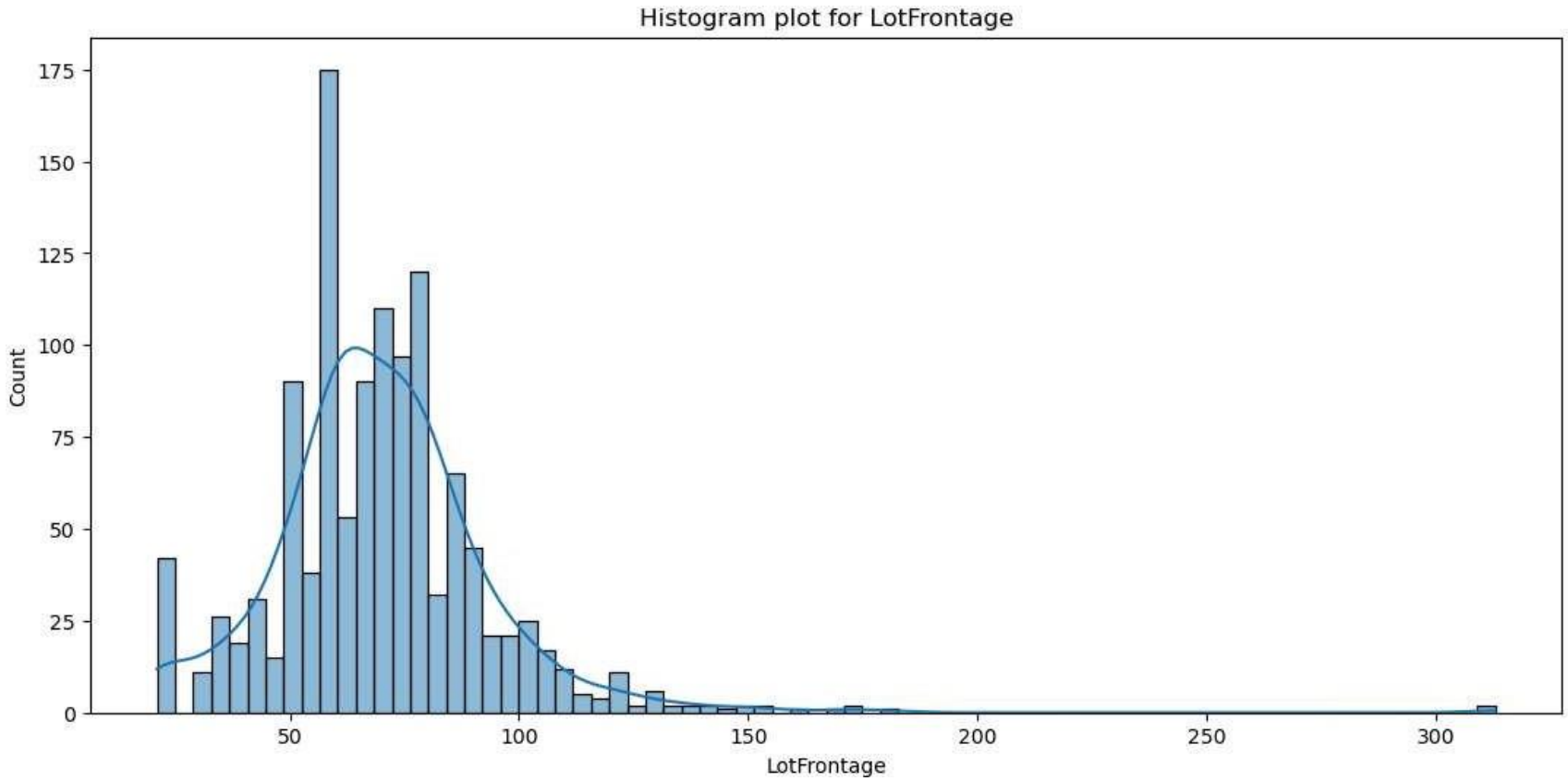


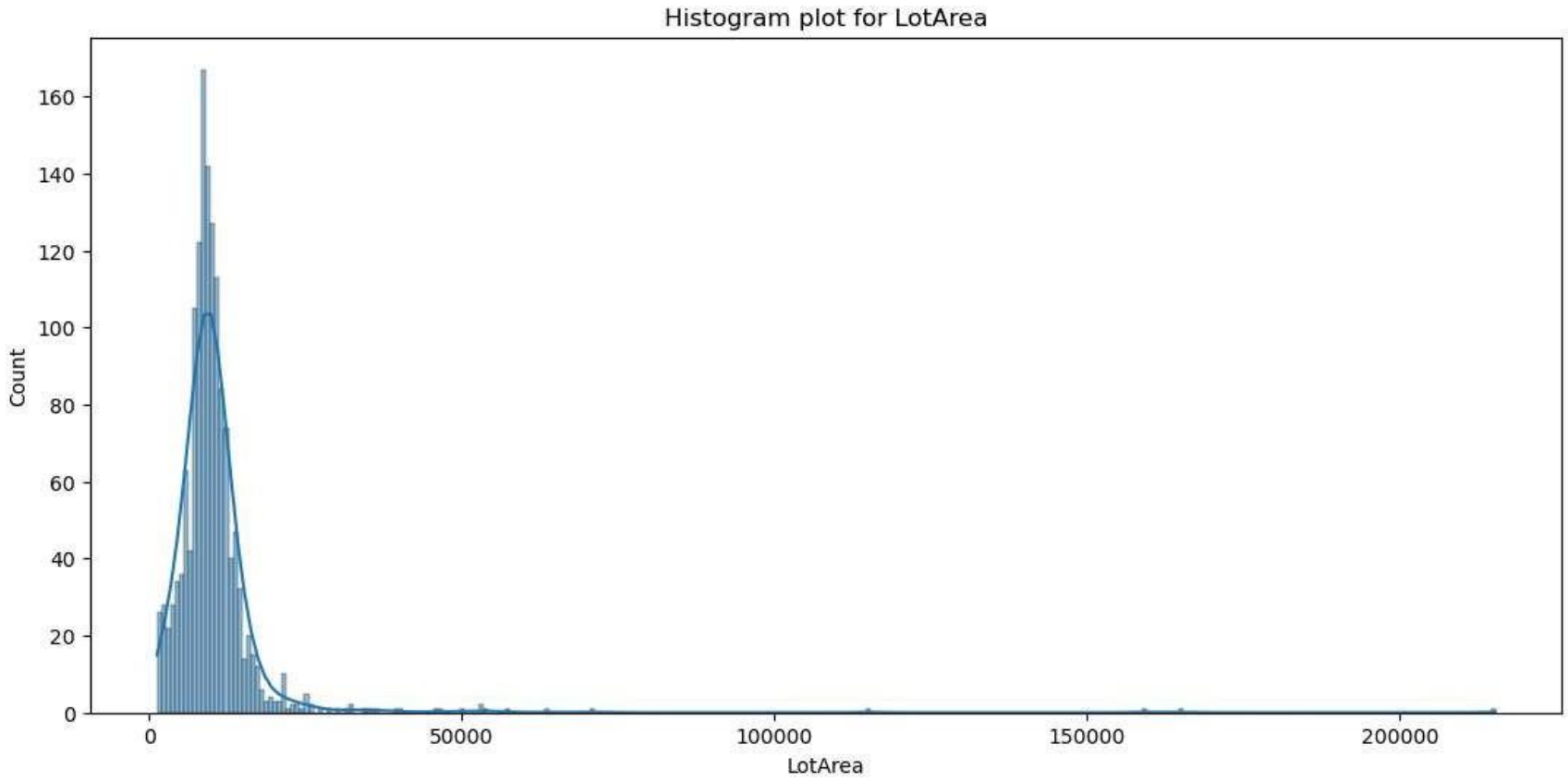
==========

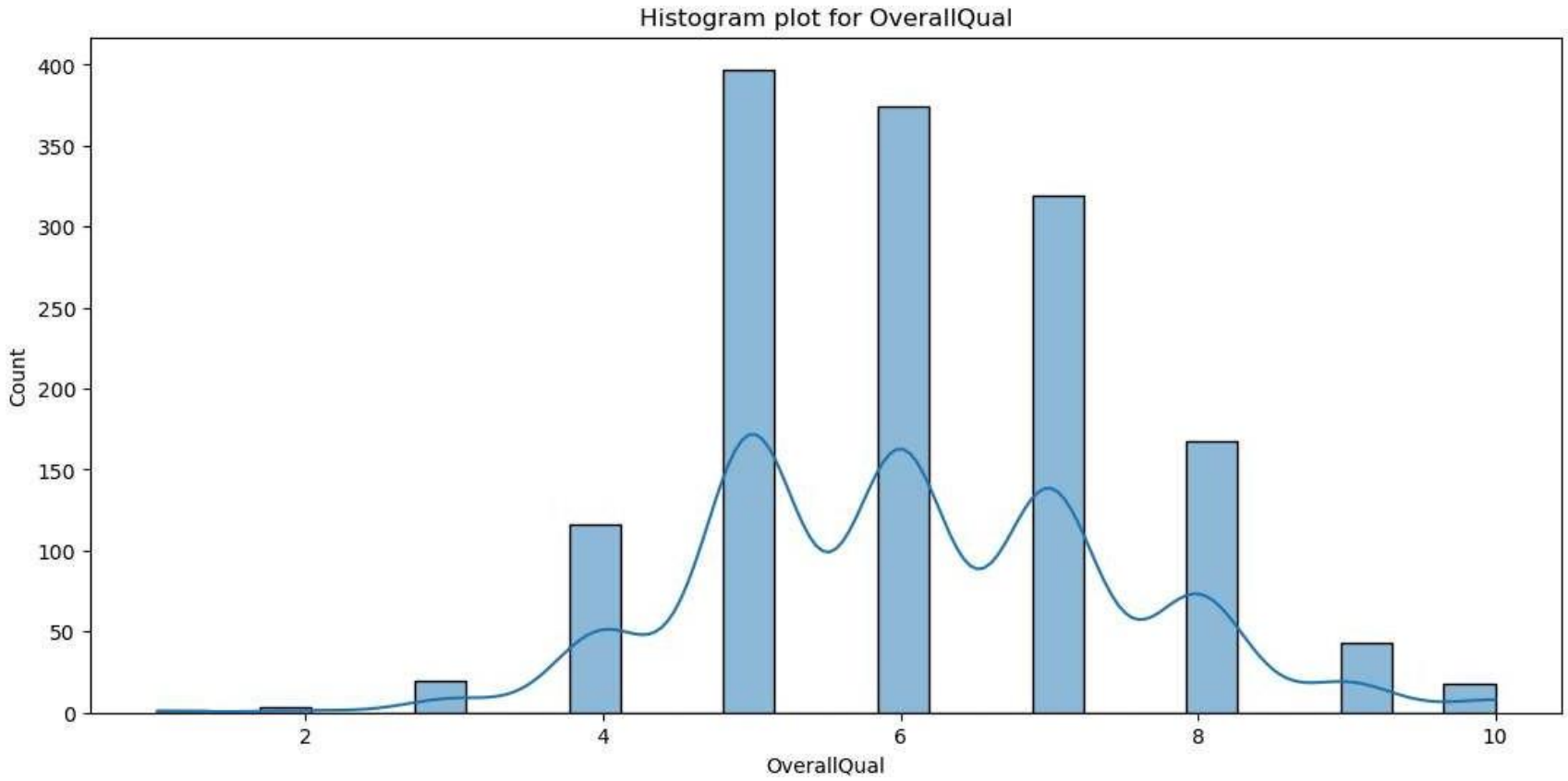
```
Contineous variable : ['Id', 'MSSubClass', 'LotFrontage', 'LotArea', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRe
modAdd', 'MasVnrArea', 'BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF',
'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath', 'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'TotRmsAbvGrd', 'F
ireplaces', 'GarageYrBlt', 'GarageCars', 'GarageArea', 'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch', 'Scre
enPorch', 'PoolArea', 'MiscVal', 'MoSold', 'YrSold', 'SalePrice'] Histogram
```

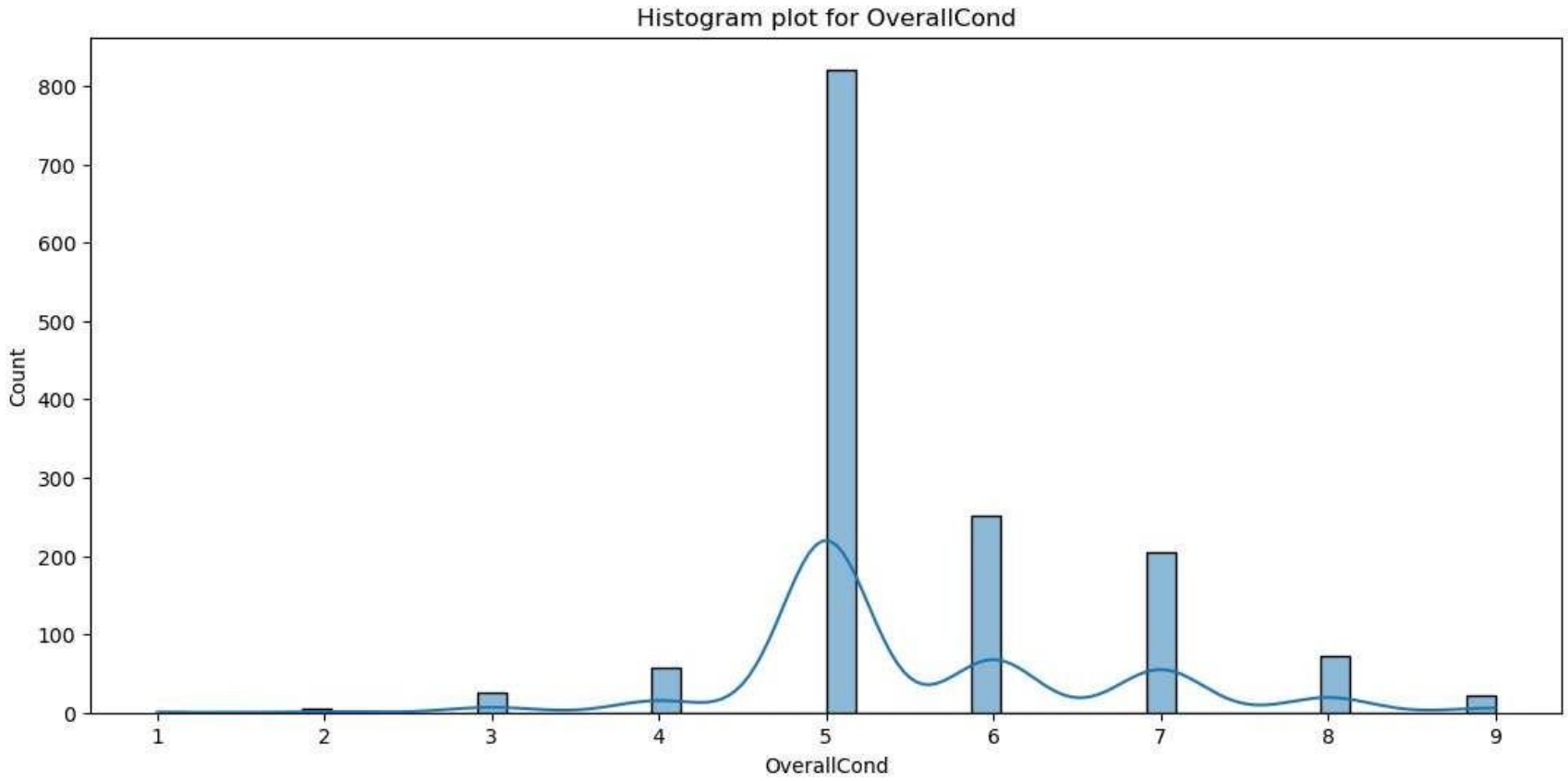

Histogram plot for Id
Count
Id
0
20
40
60
80
100
120
0
200
400
600
800
1000
1200
1400

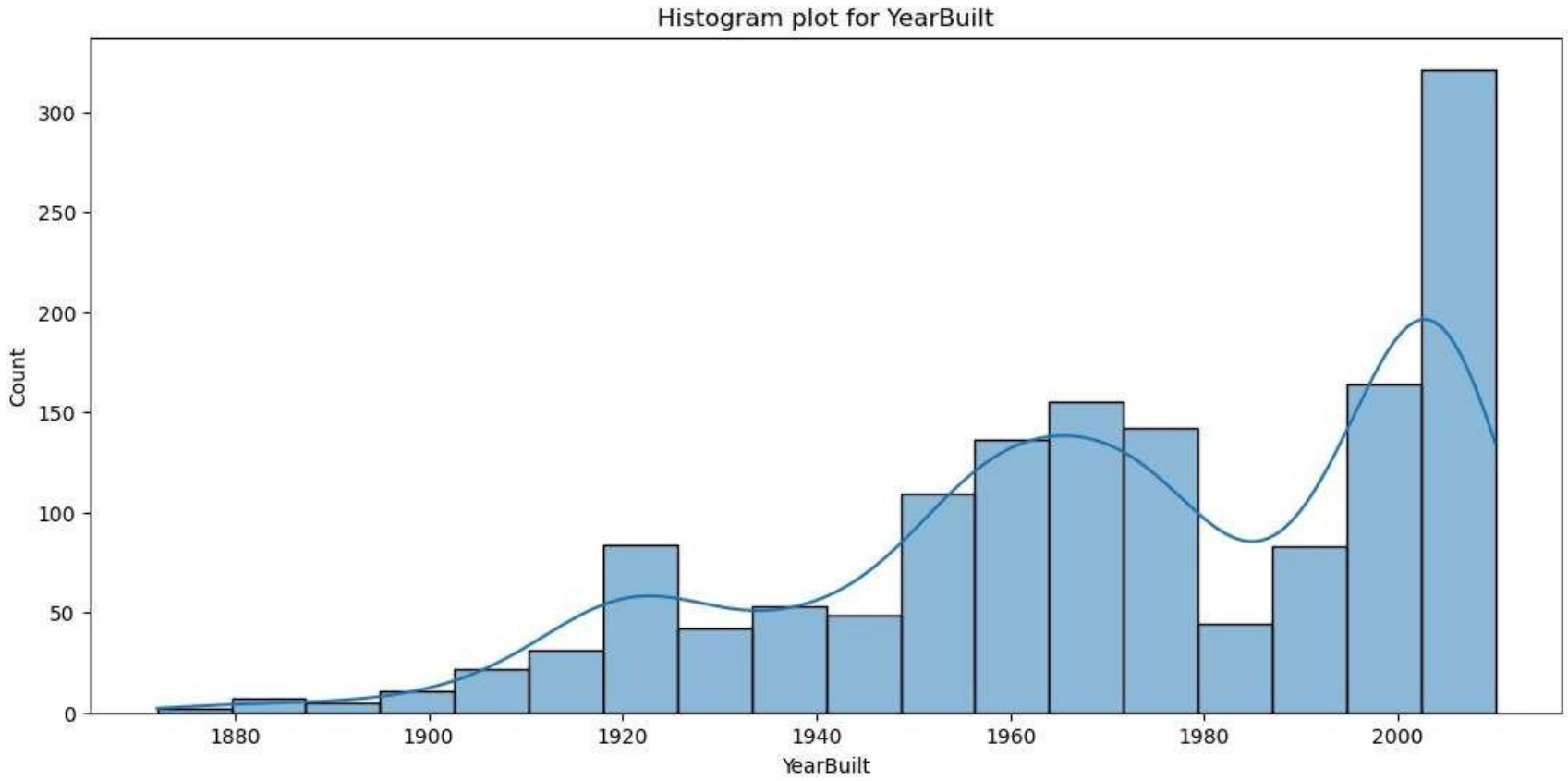

Histogram plot for MSSubClass
Count
MSSubClass
0
100
200
300
400
500
25
50
75
100
125
150
175

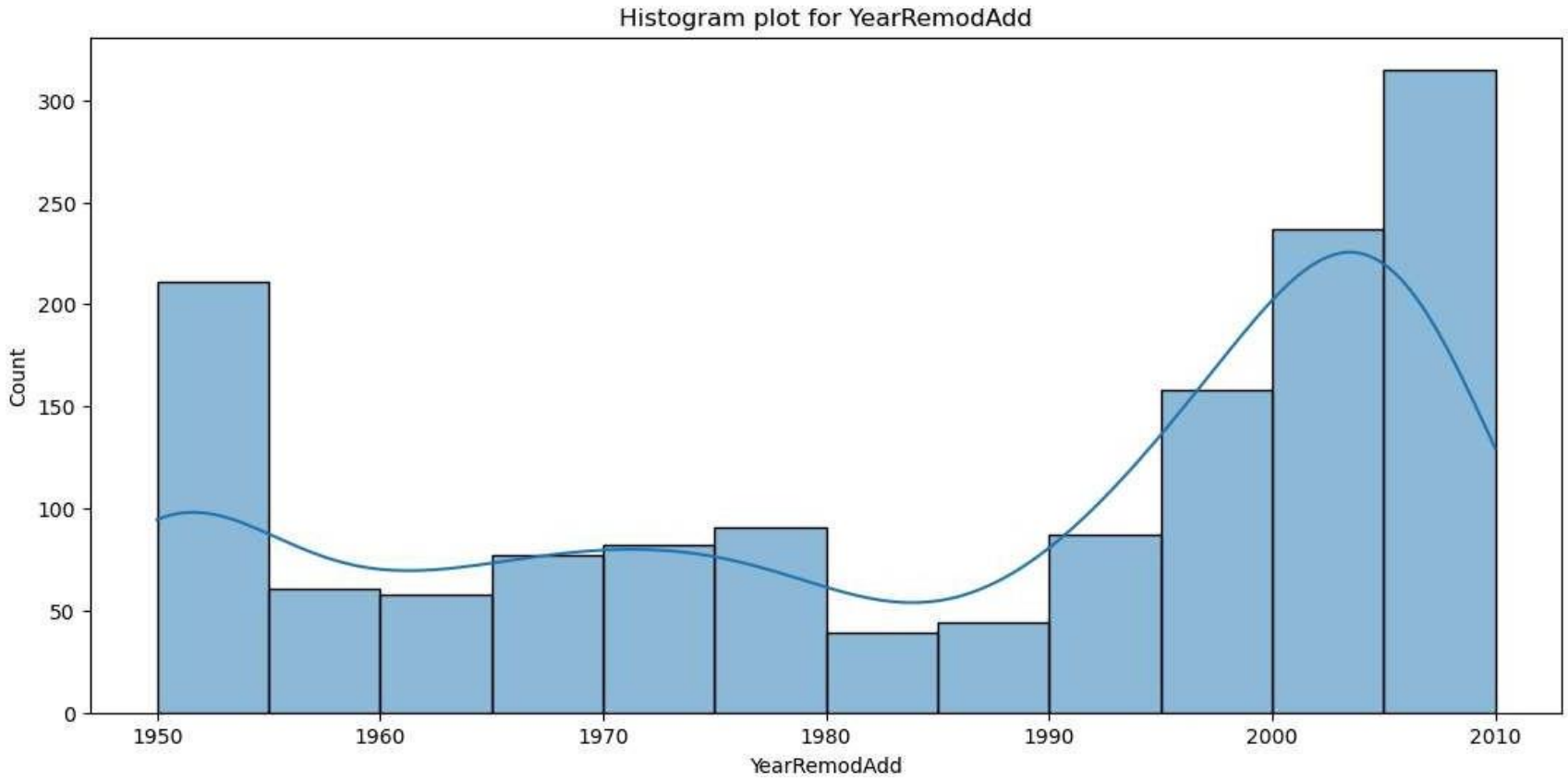

Histogram plot for LotFrontage
Count
LotFrontage
0
25
50
75
100
125
150
175
50
100
150
200
250
300

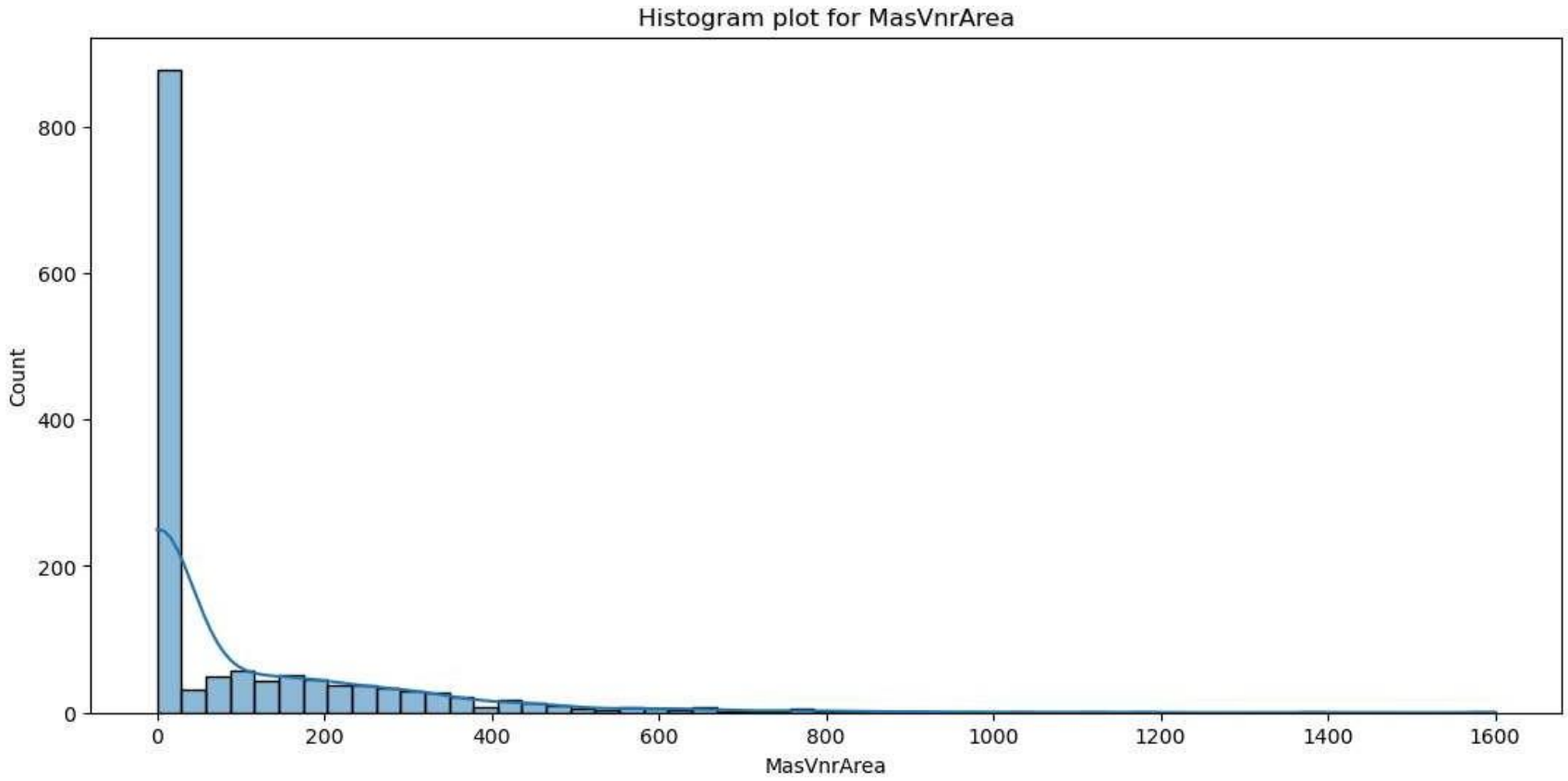

Histogram plot for LotArea
Count
LotArea
0
20
40
60
80
100
120
140
160
0
50000
100000
150000
200000

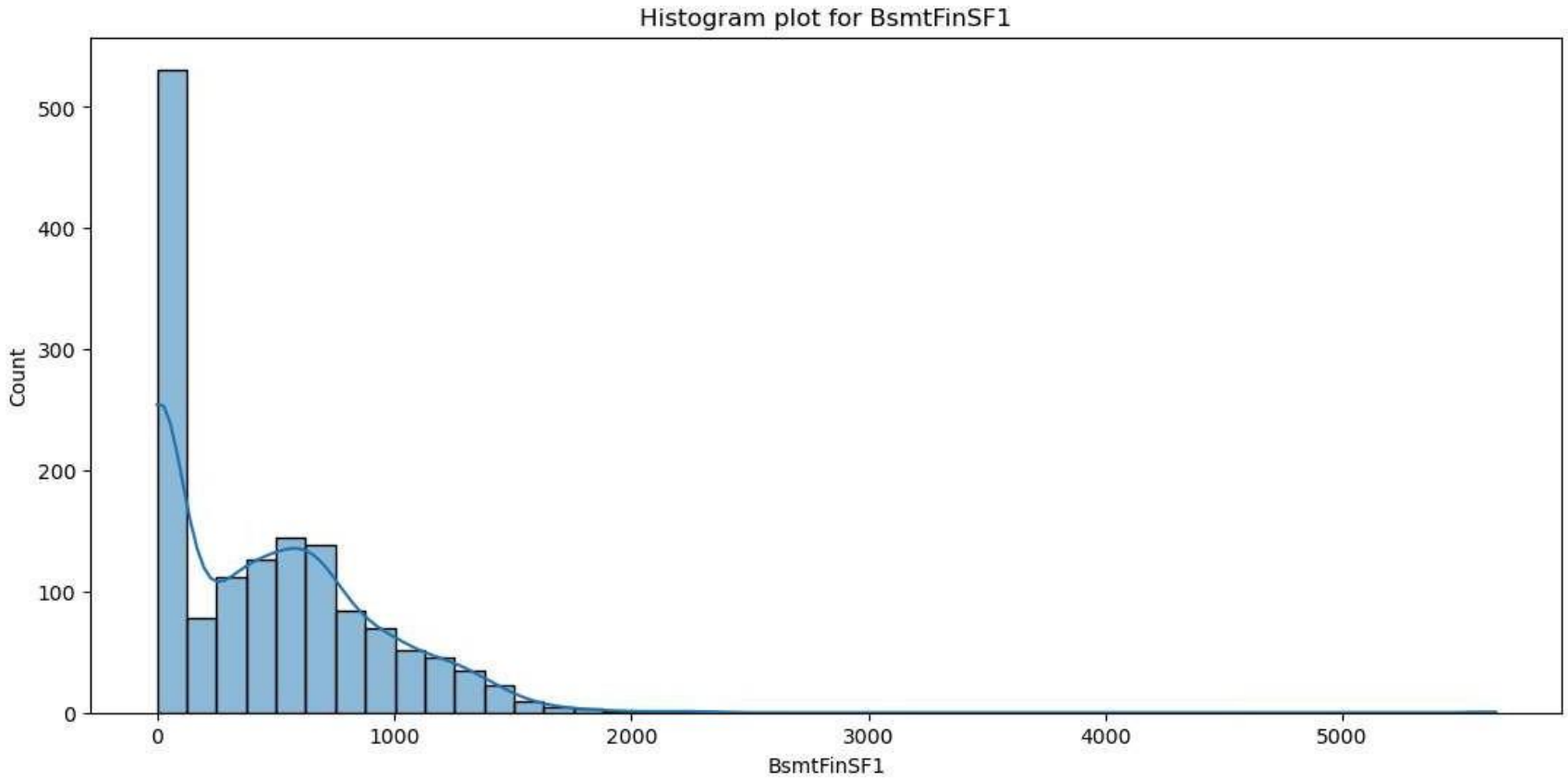

Histogram plot for OverallQual
Count
OverallQual
0
50
100
150
200
250
300
350
400
2
4
6
8
10

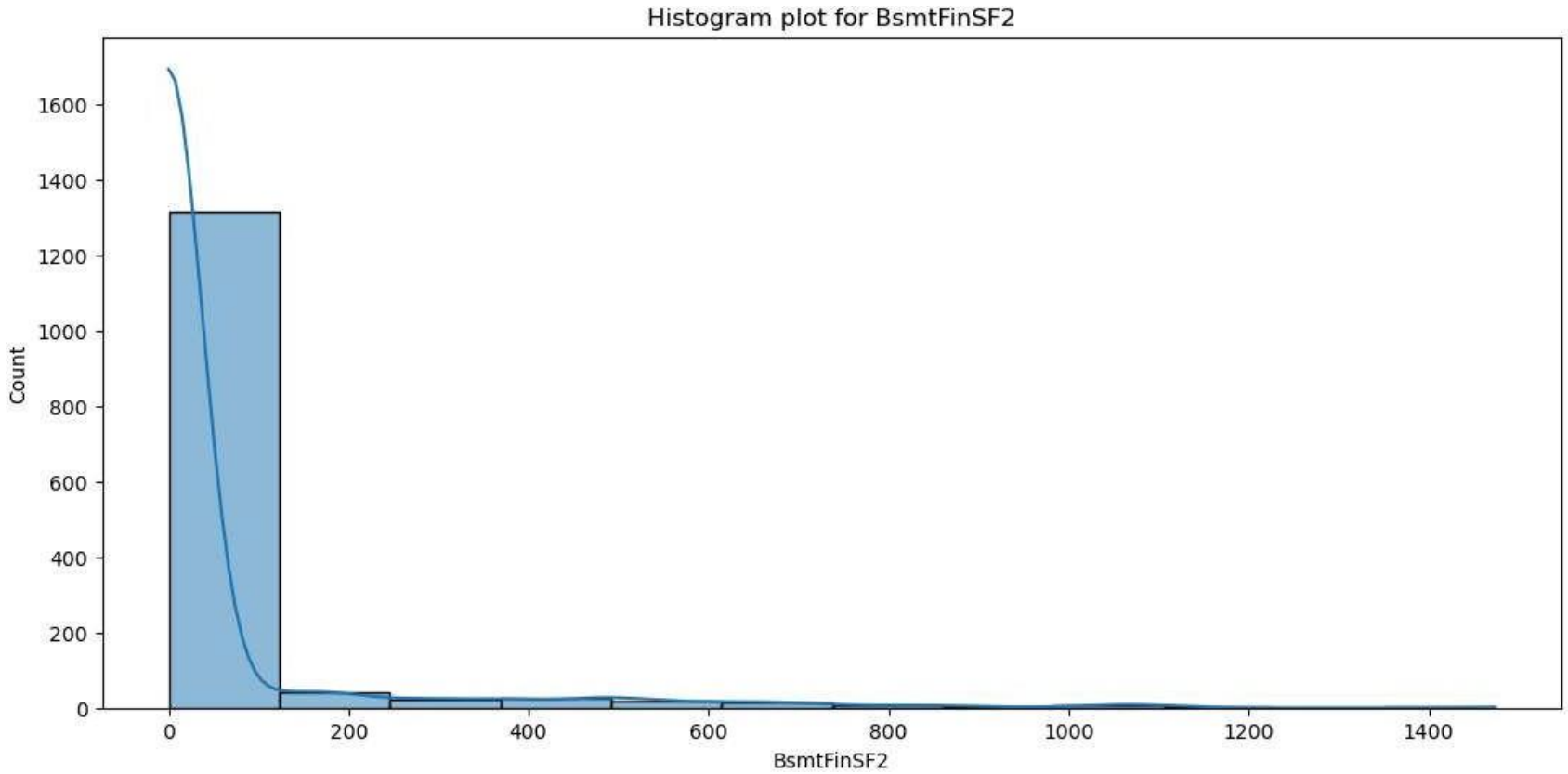

Histogram plot for OverallCond
Count
OverallCond
0
100
200
300
400
500
600
700
800
1
2
3
4
5
6
7
8
9

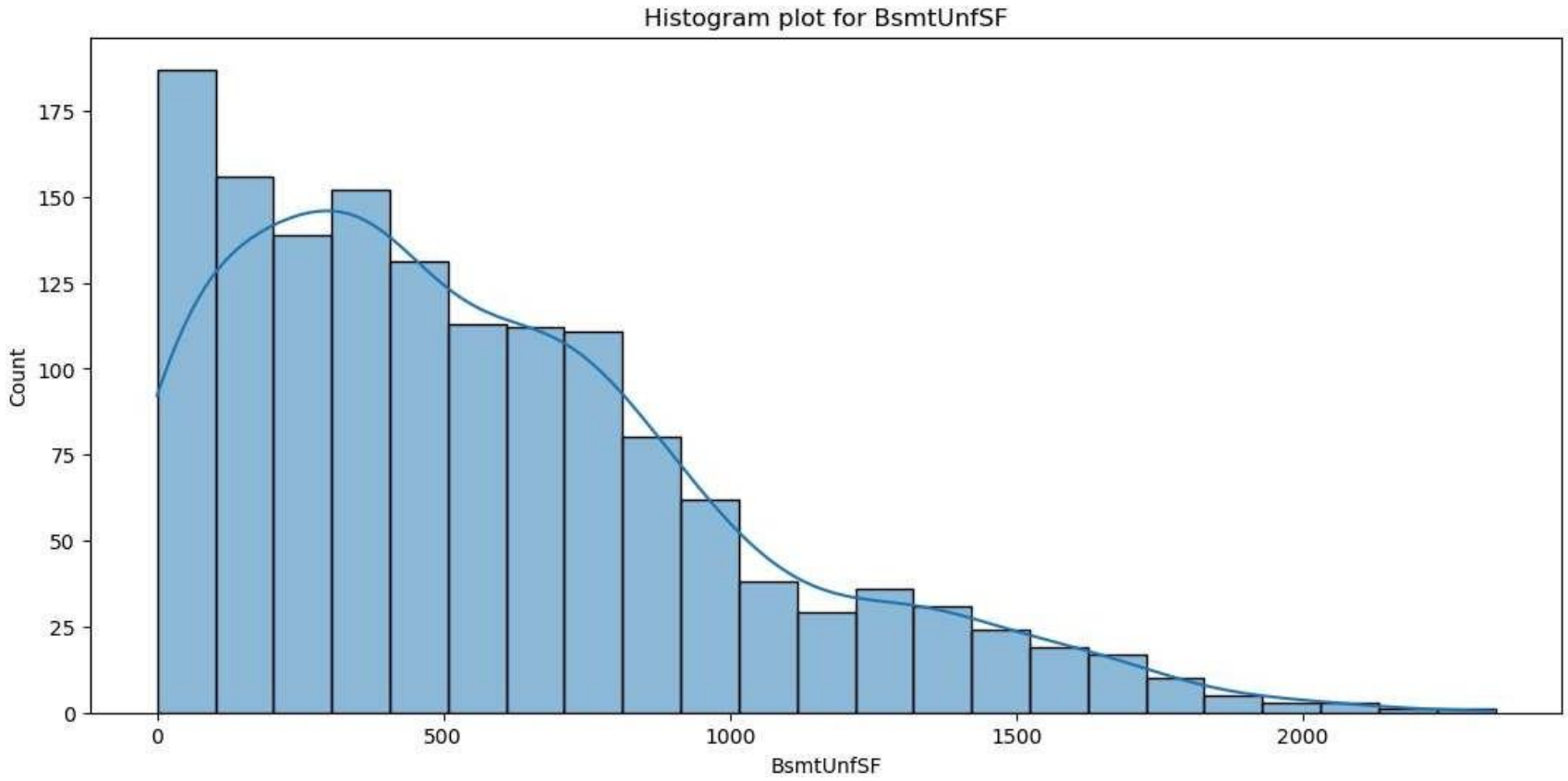

Histogram plot for YearBuilt
Count
YearBuilt
0
50
100
150
200
250
300
1880
1900
1920
1940
1960
1980
2000

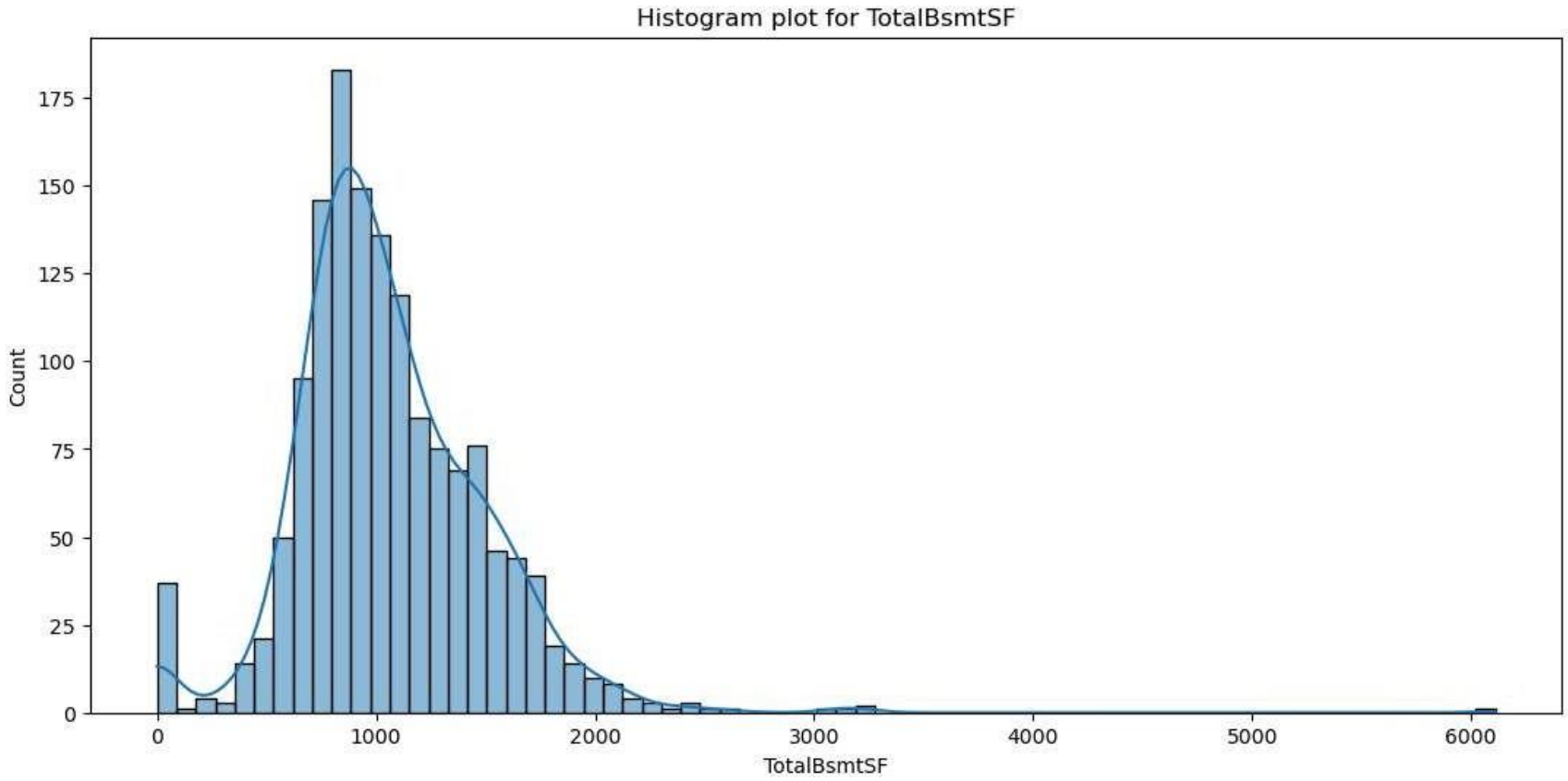

Histogram plot for YearRemodAdd
Count
YearRemodAdd
0
50
100
150
200
250
300
1950
1960
1970
1980
1990
2000
2010

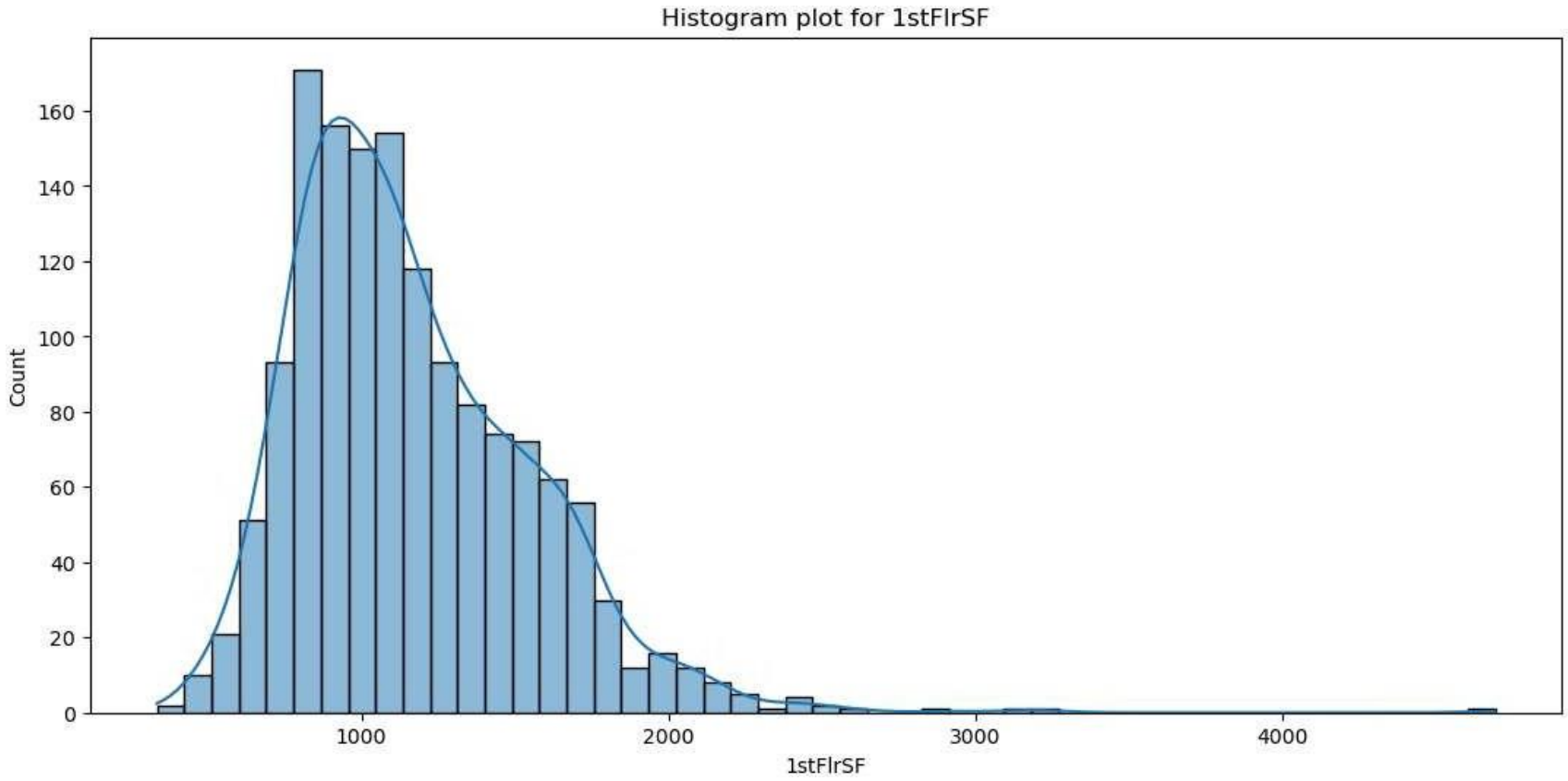

Histogram plot for MasVnrArea
Count
MasVnrArea
0
200
400
600
800
1000
1200
1400
1600

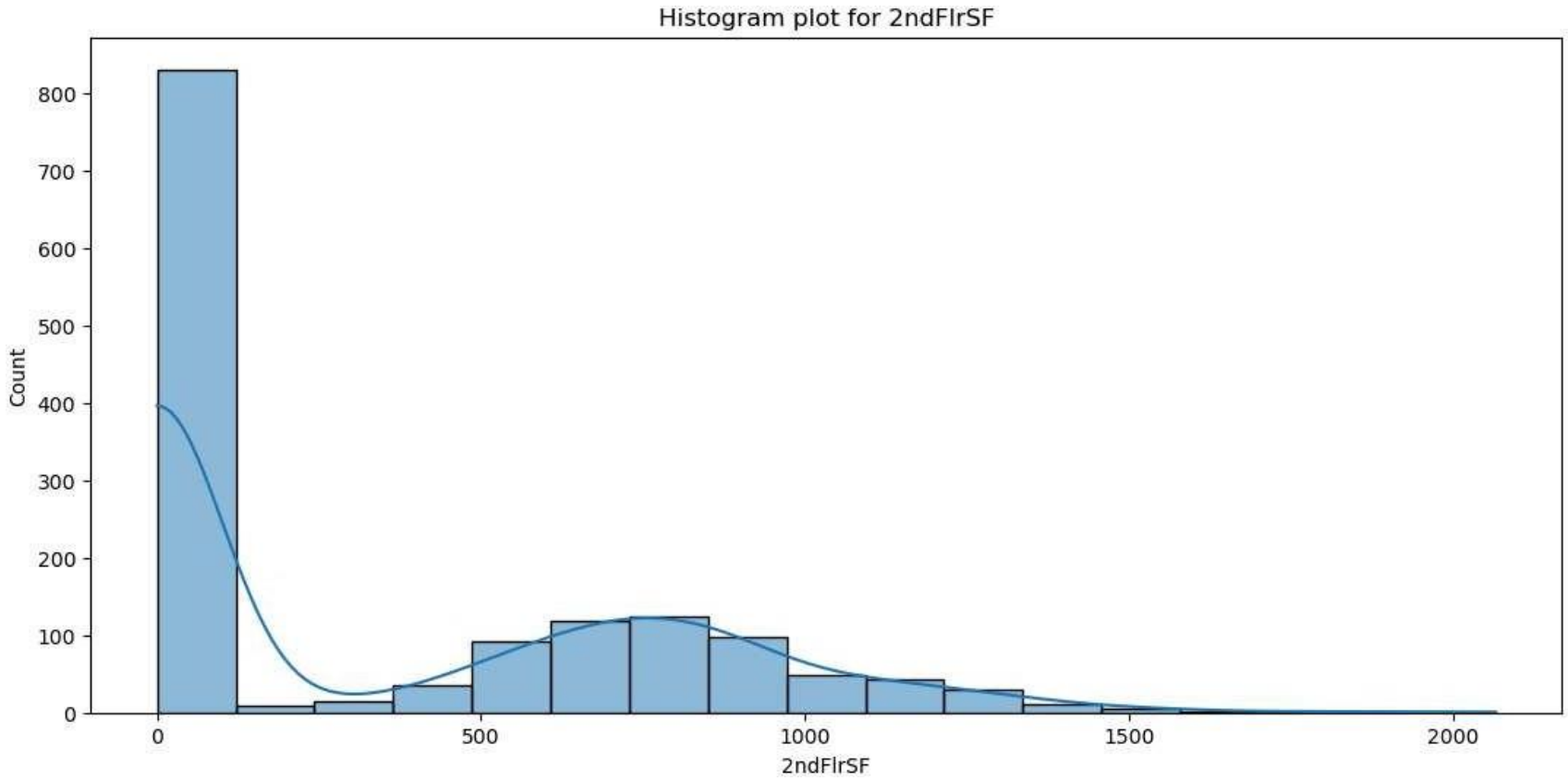

Histogram plot for BsmtFinSF1
Count
BsmtFinSF1
0
100
200
300
400
500
0
1000
2000
3000
4000
5000

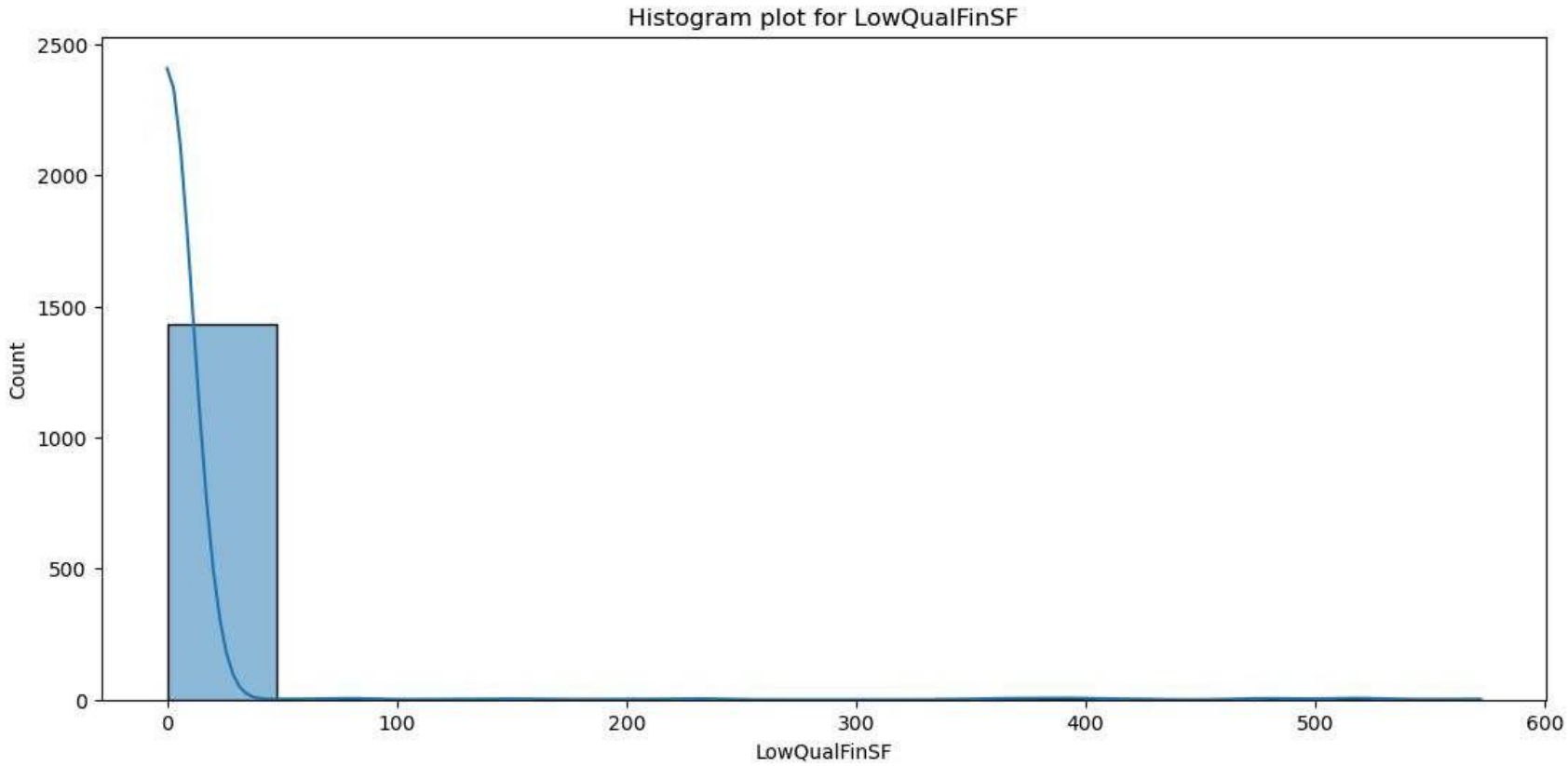

Histogram plot for BsmtFinSF2
Count
BsmtFinSF2
0
200
400
600
800
1000
1200
1400
1600

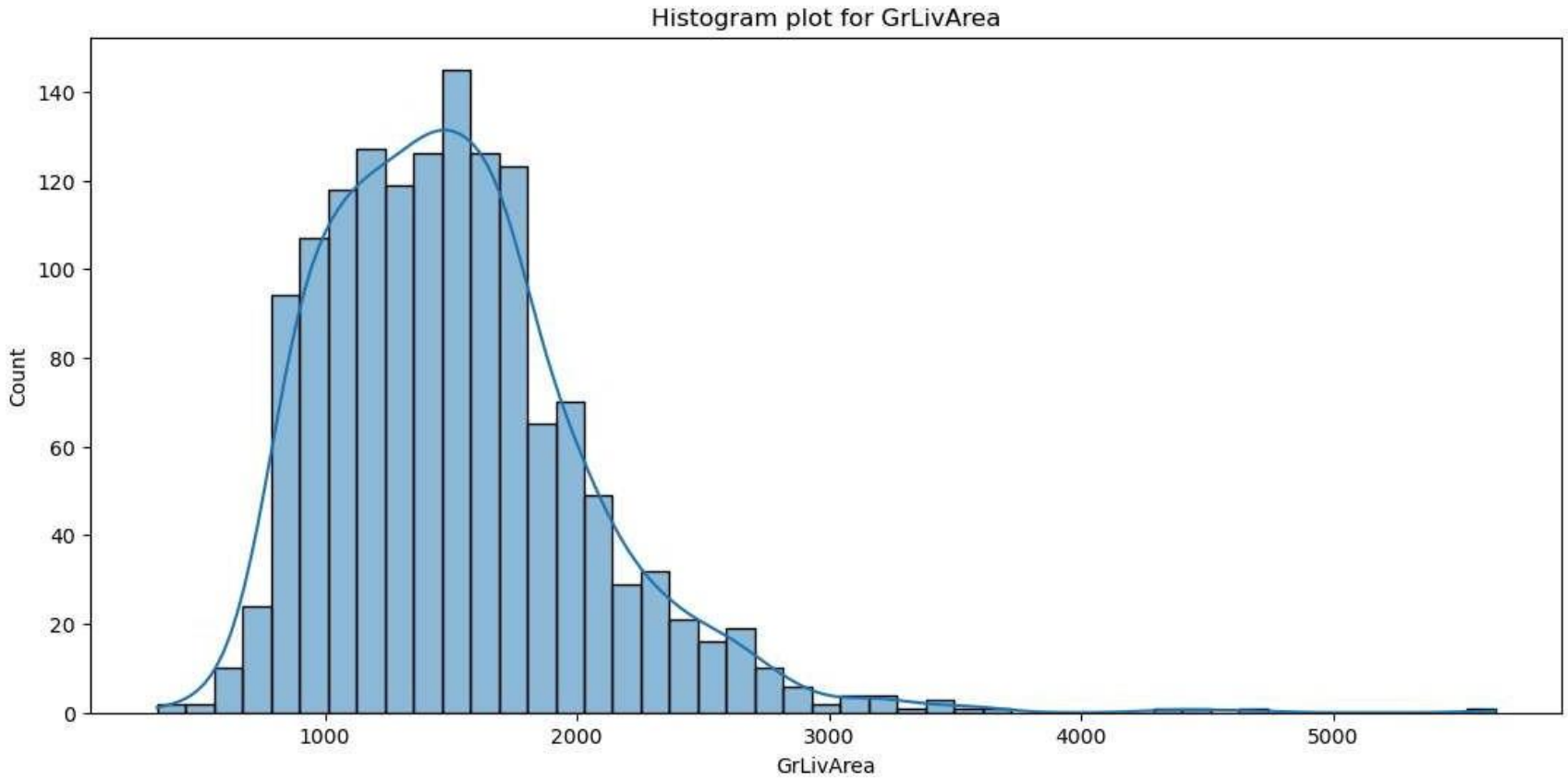

Histogram plot for BsmtUnfSF
Count
BsmtUnfSF
0
25
50
75
100
125
150
175
0
500
1000
1500
2000

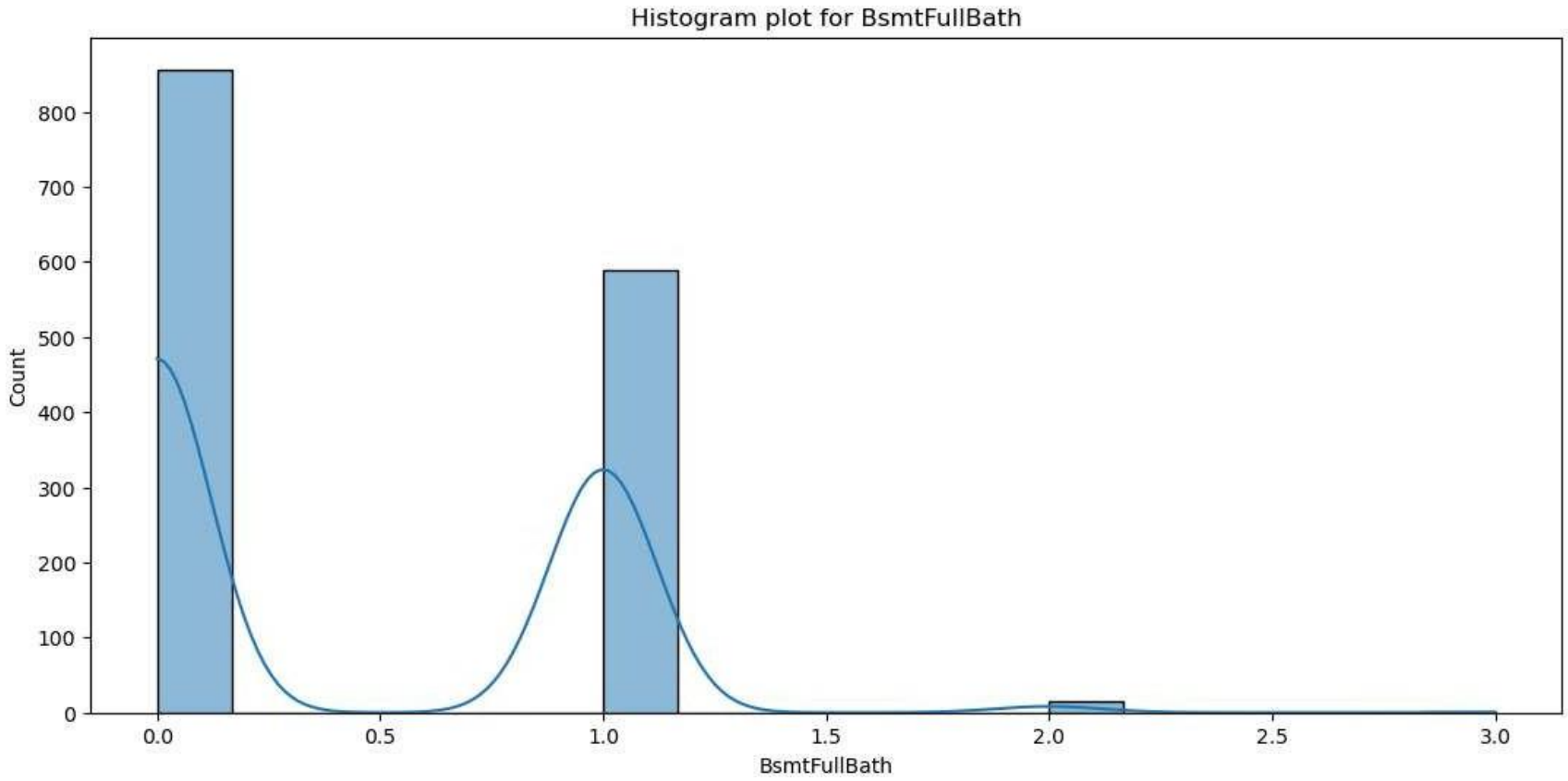

Histogram plot for TotalBsmtSF
Count
TotalBsmtSF
0
25
50
75
100
125
150
175
0
1000
2000
3000
4000
5000
6000

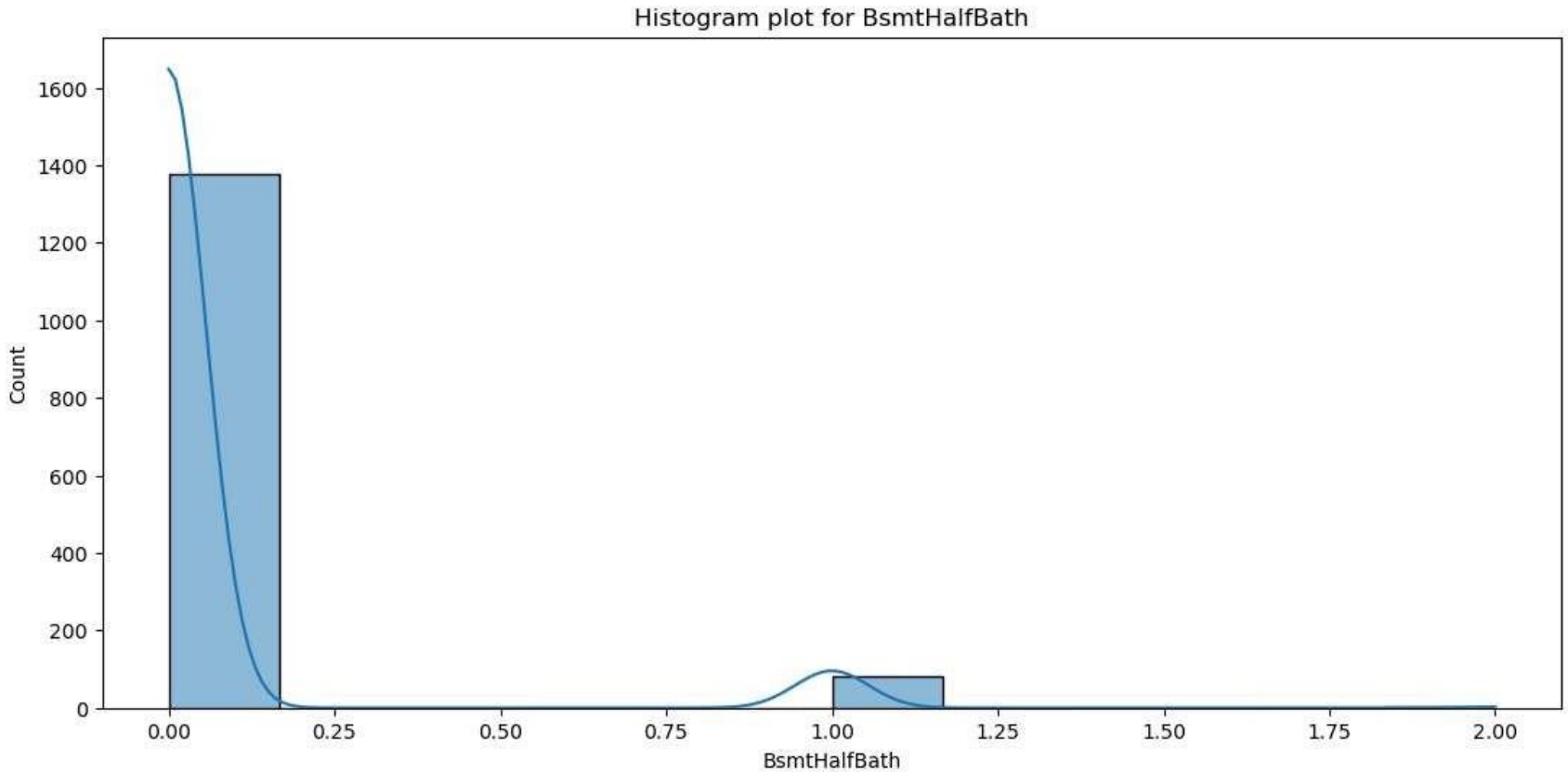

Histogram plot for 1stFlrSF
Count
0
20
40
60
80
100
120
140
160
1000
2000
3000
4000
1stFlrSF

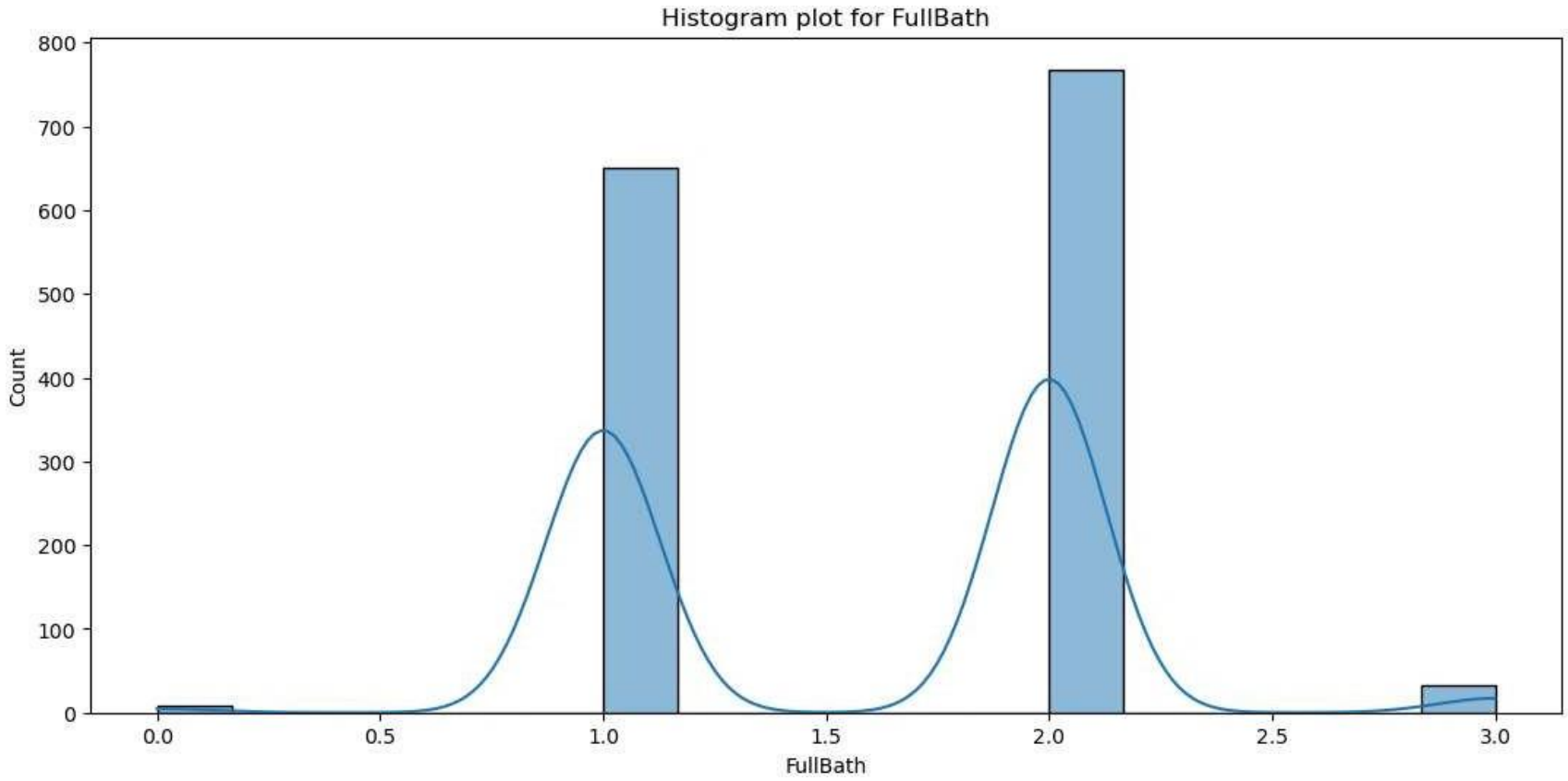

Histogram plot for 2ndFlrSF
Count
2ndFlrSF
0
100
200
300
400
500
600
700
800
0
500
1000
1500
2000

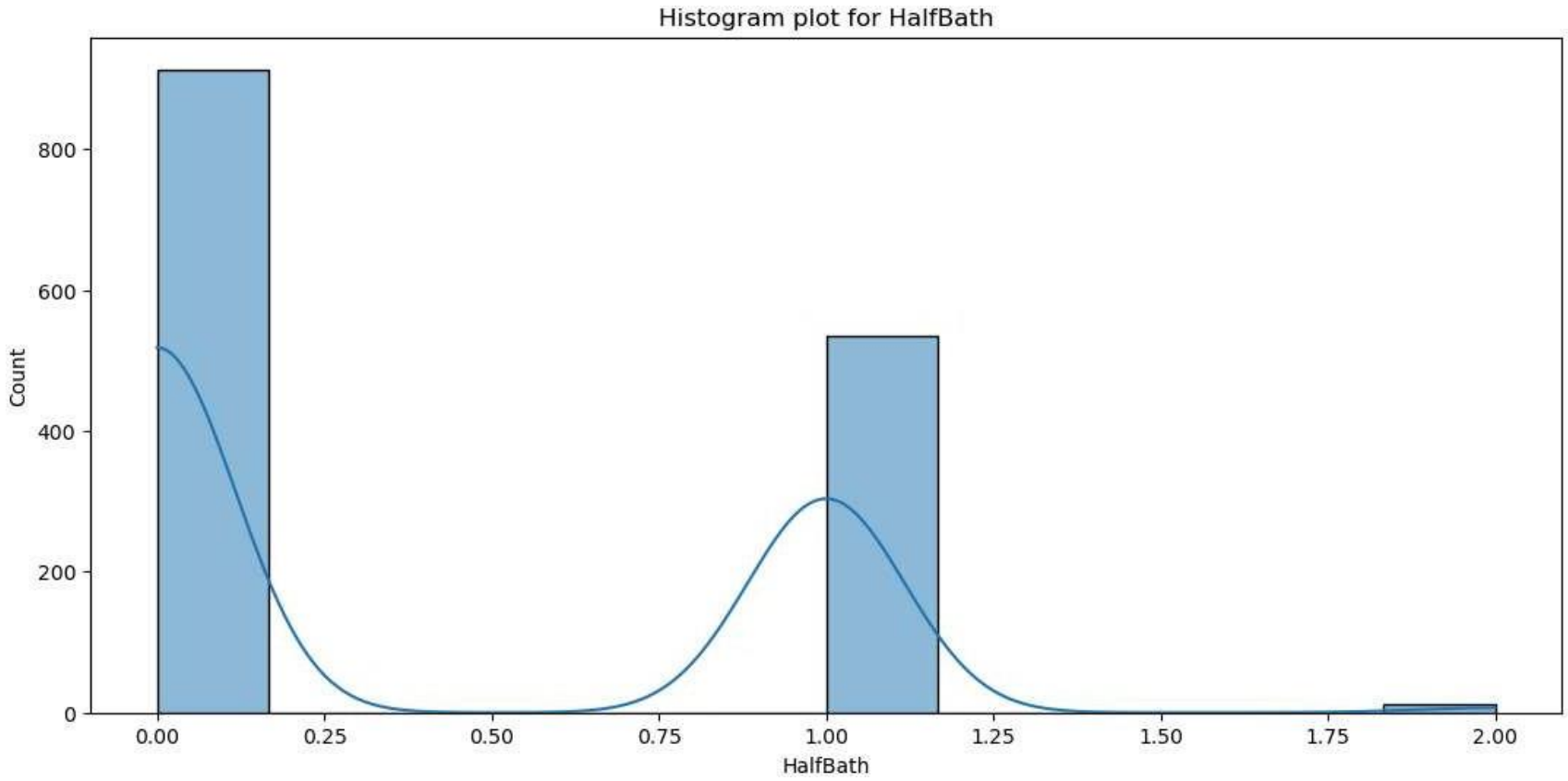

Histogram plot for LowQualFinSF
Count
LowQualFinSF
0
500
1000
1500
2000
2500
0
100
200
300
400
500
600

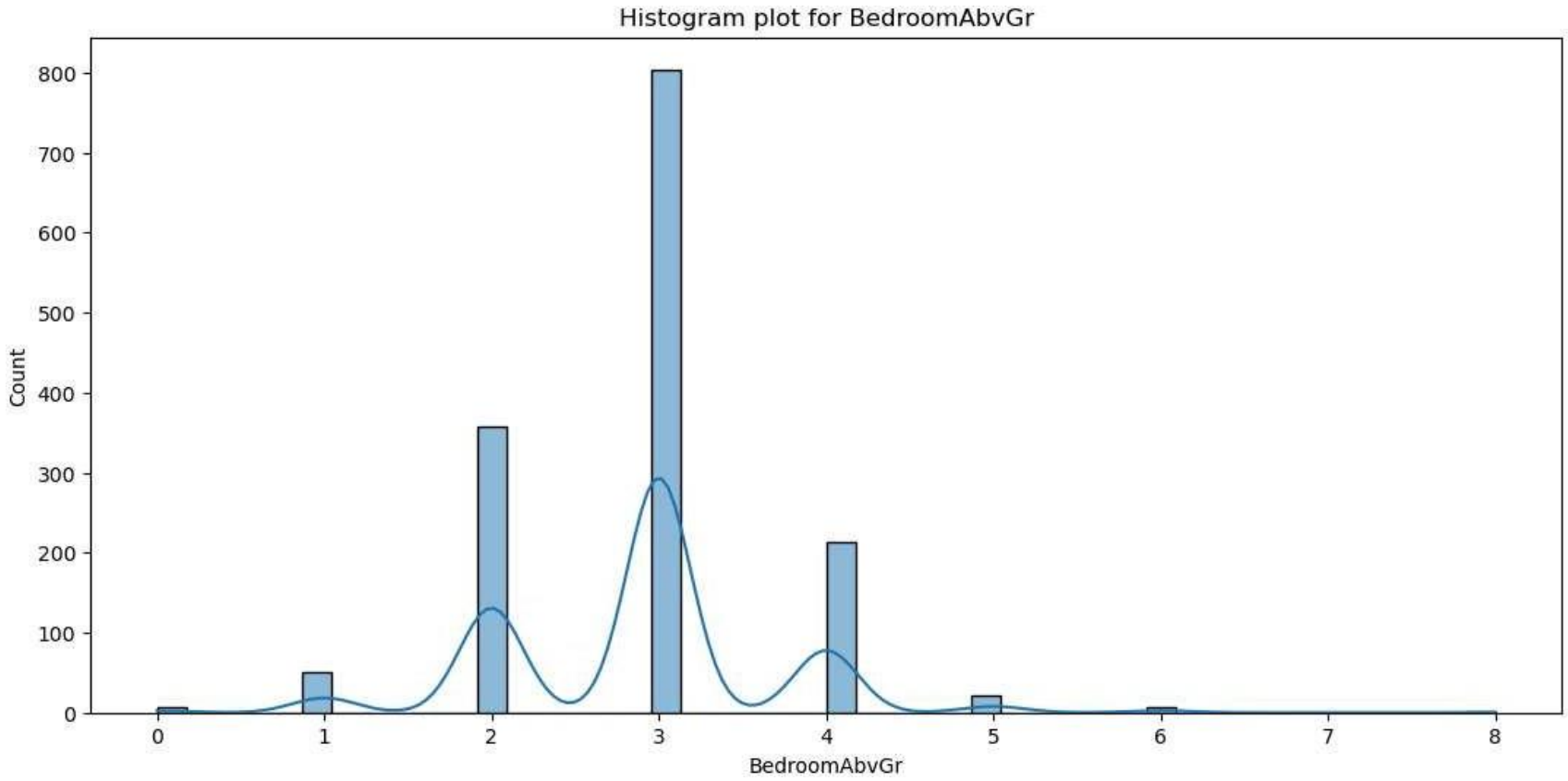

Histogram plot for GrLivArea
Count
GrLivArea
0
20
40
60
80
100
120
140
1000
2000
3000
4000
5000

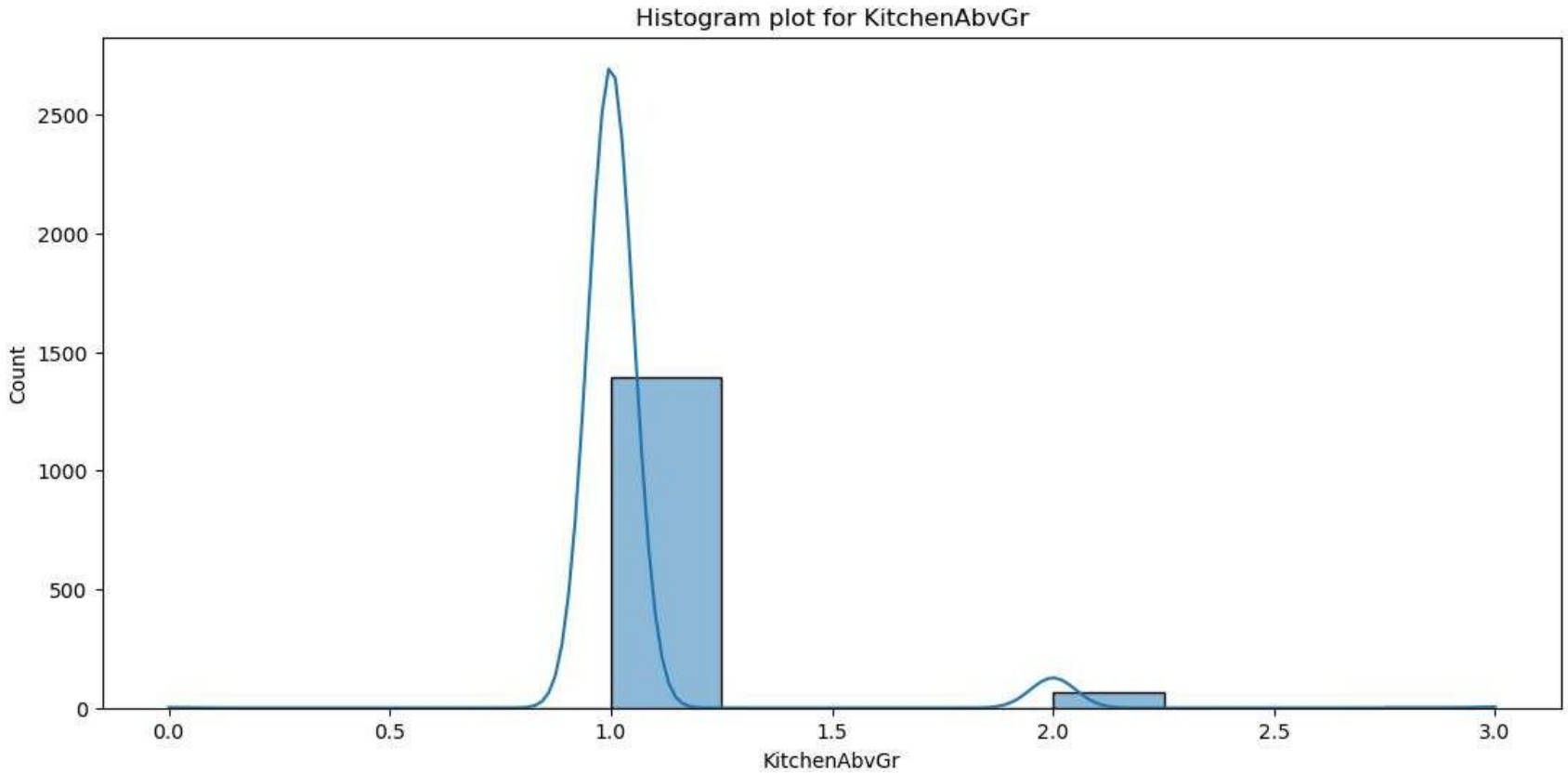

Histogram plot for BsmtFullBath
Count
BsmtFullBath
0
100
200
300
400
500
600
700
800
0.0
0.5
1.0
1.5
2.0
2.5
3.0

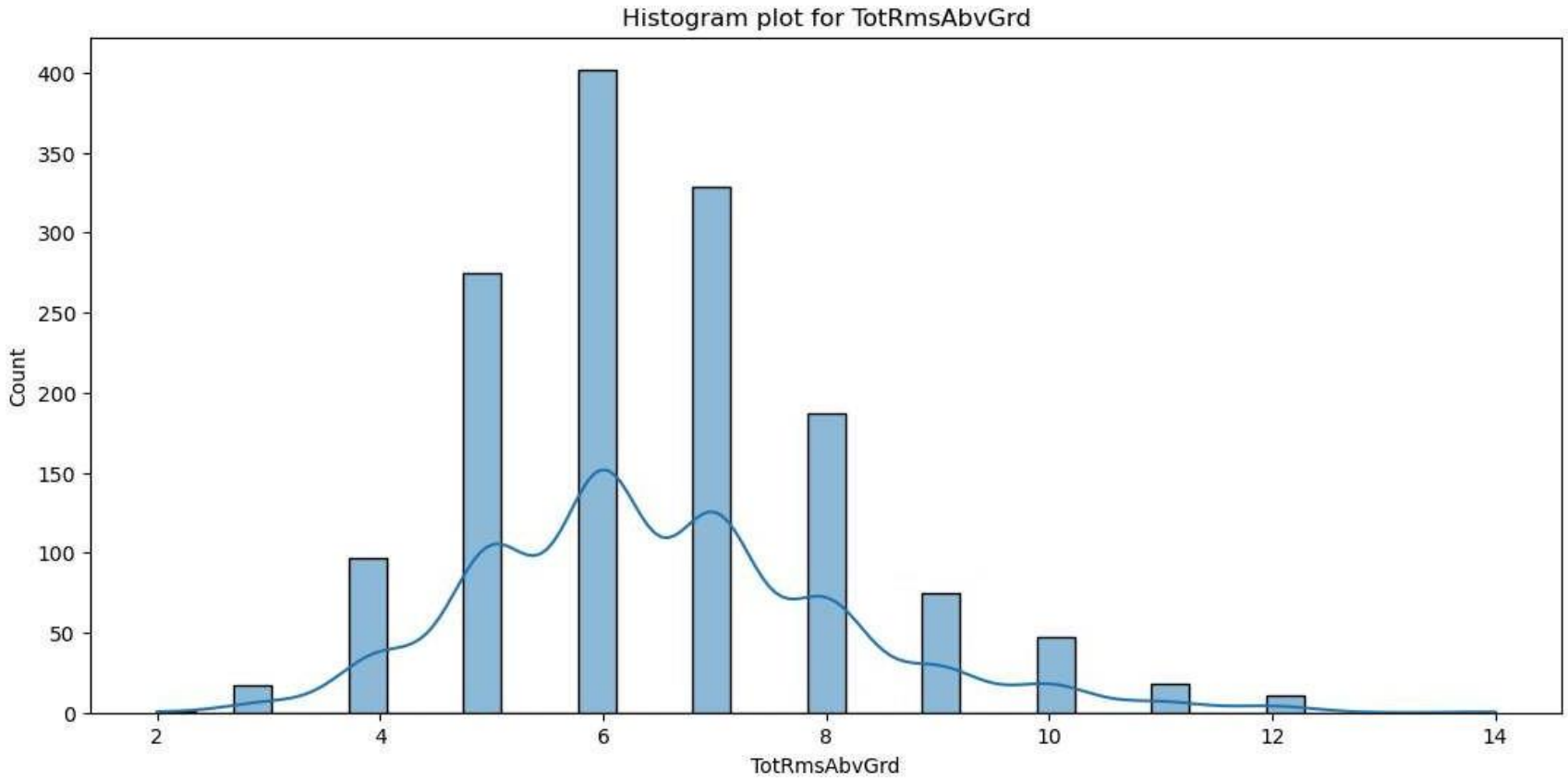

Histogram plot for BsmtHalfBath
Count
BsmtHalfBath
0
200
400
600
800
1000
1200
1400
1600
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00

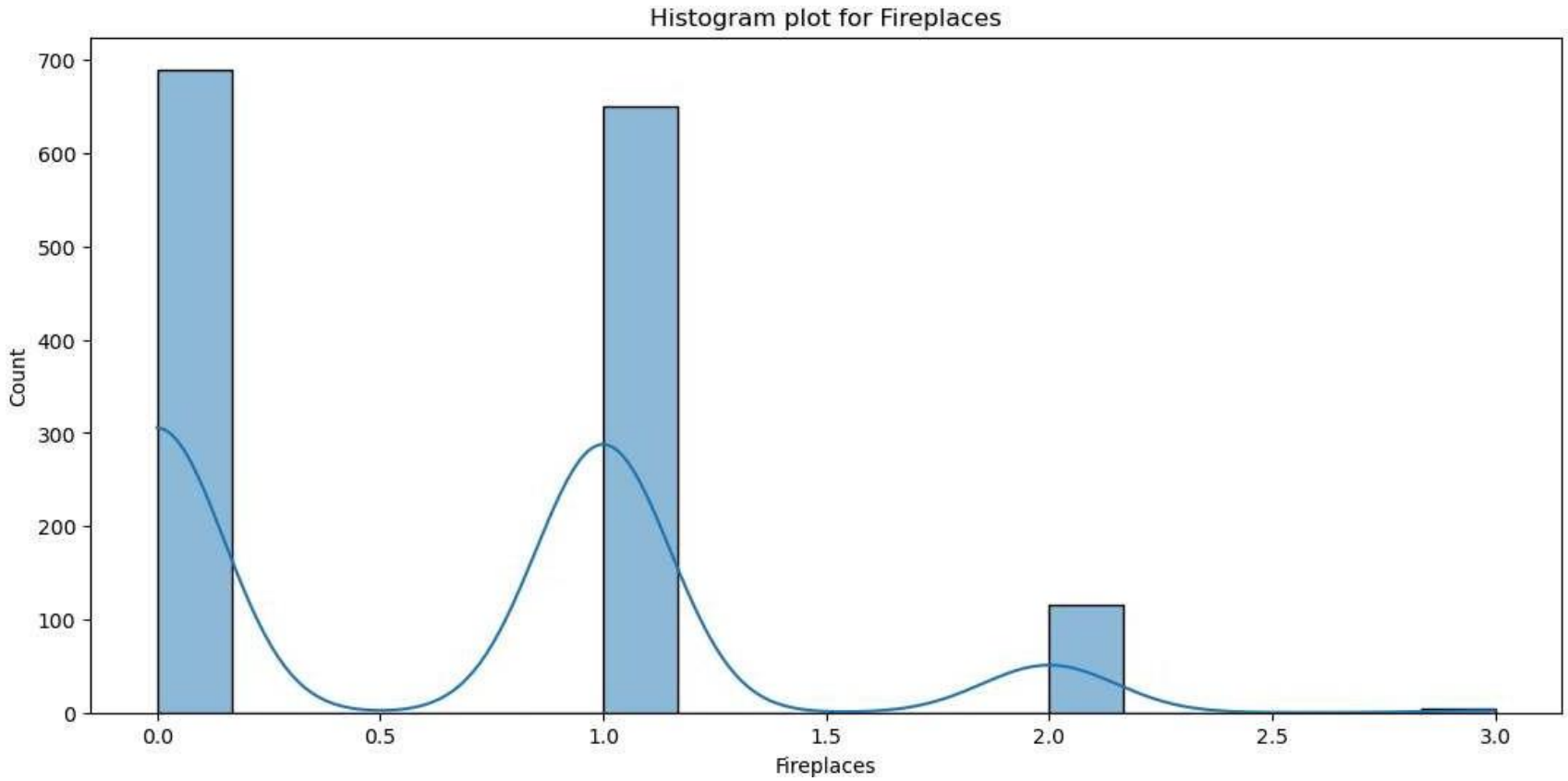

Histogram plot for FullBath
Count
FullBath
0
100
200
300
400
500
600
700
800
0.0
0.5
1.0
1.5
2.0
2.5
3.0

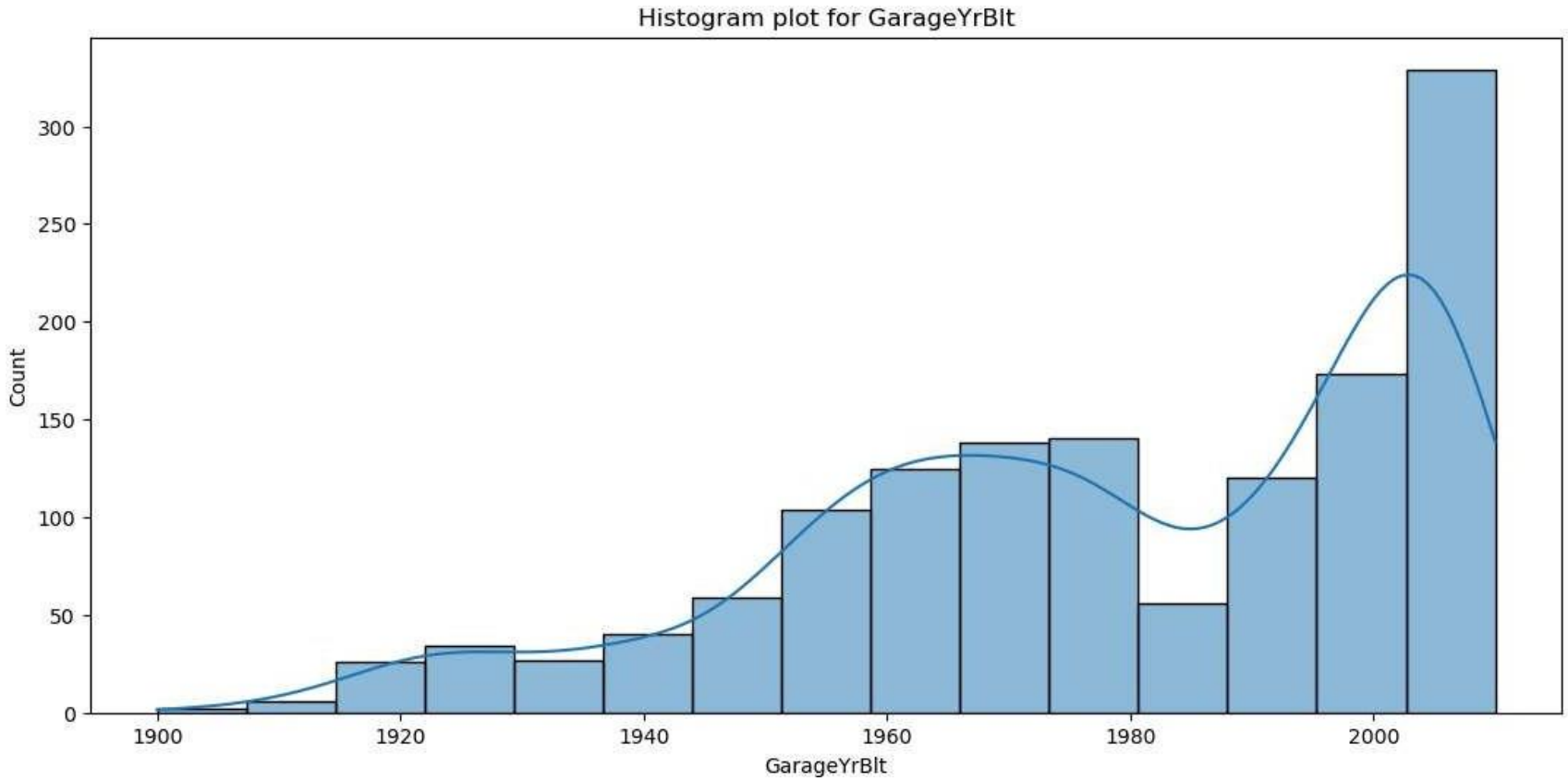

Histogram plot for HalfBath
Count
HalfBath
0
200
400
600
800
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00

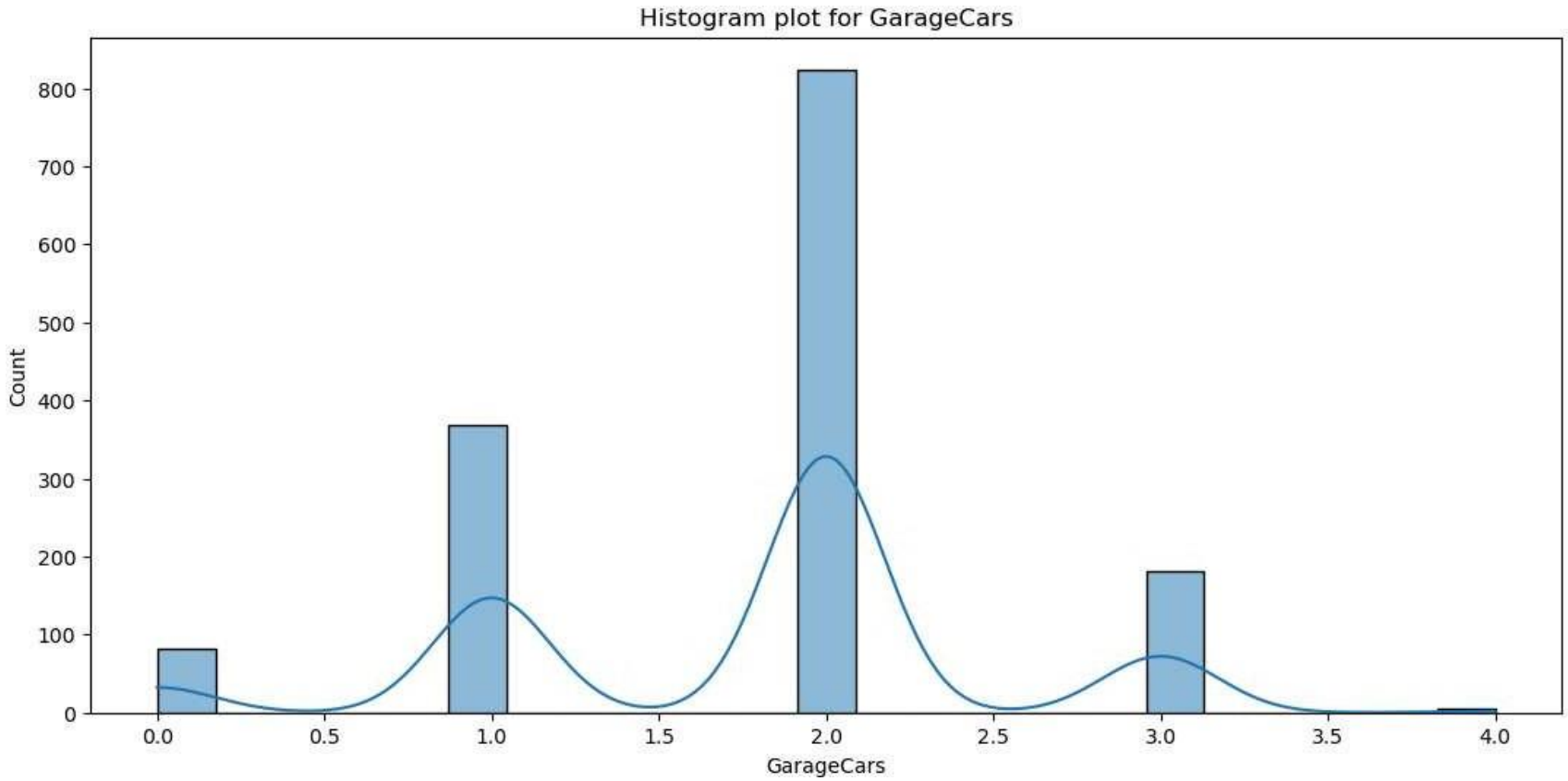

Histogram plot for BedroomAbvGr
Count
BedroomAbvGr
0
100
200
300
400
500
600
700
800
0
1
2
3
4
5
6
7
8

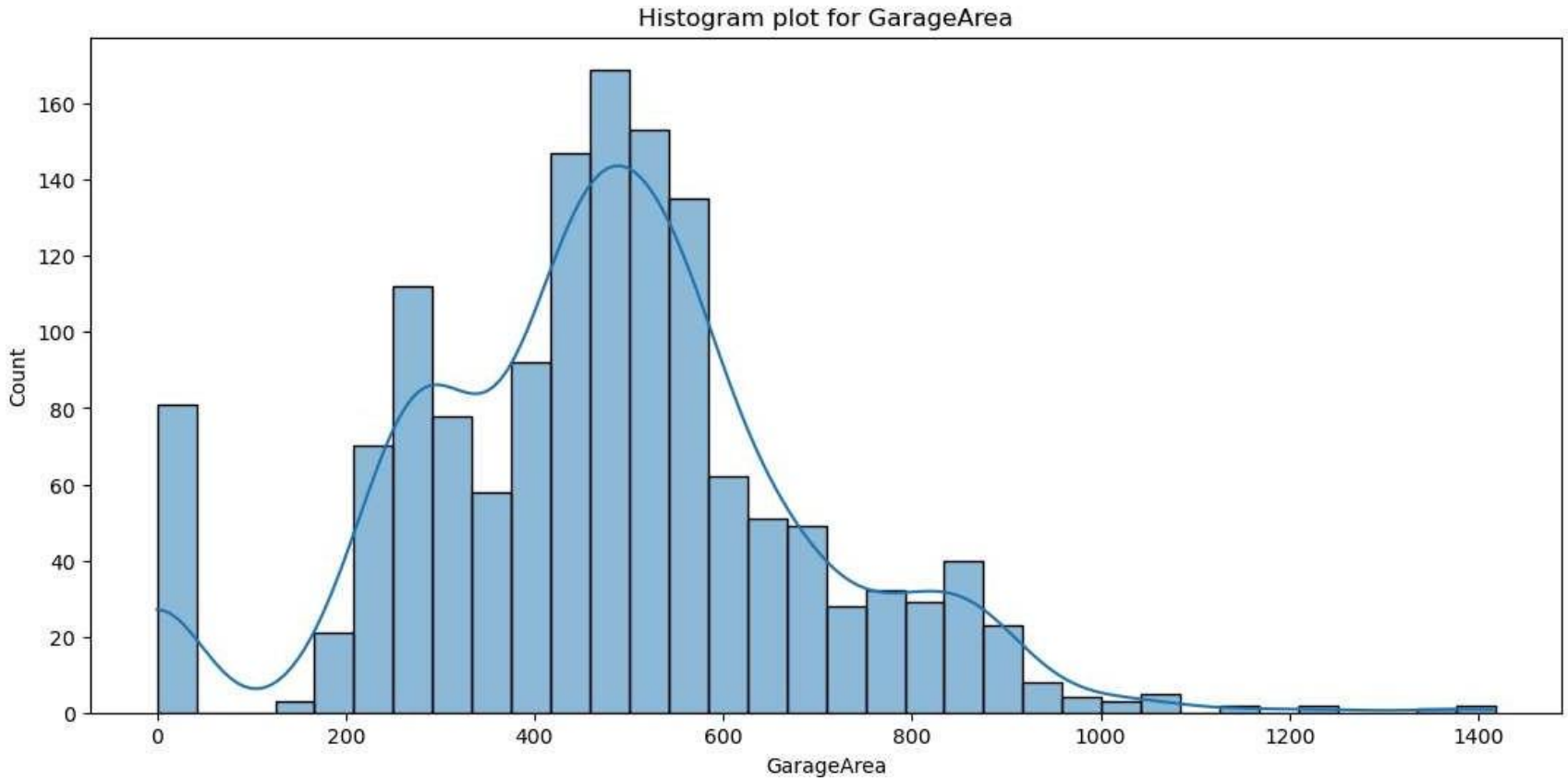

Histogram plot for KitchenAbvGr
Count
2500
2000
1500
1000
500
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
KitchenAbvGr

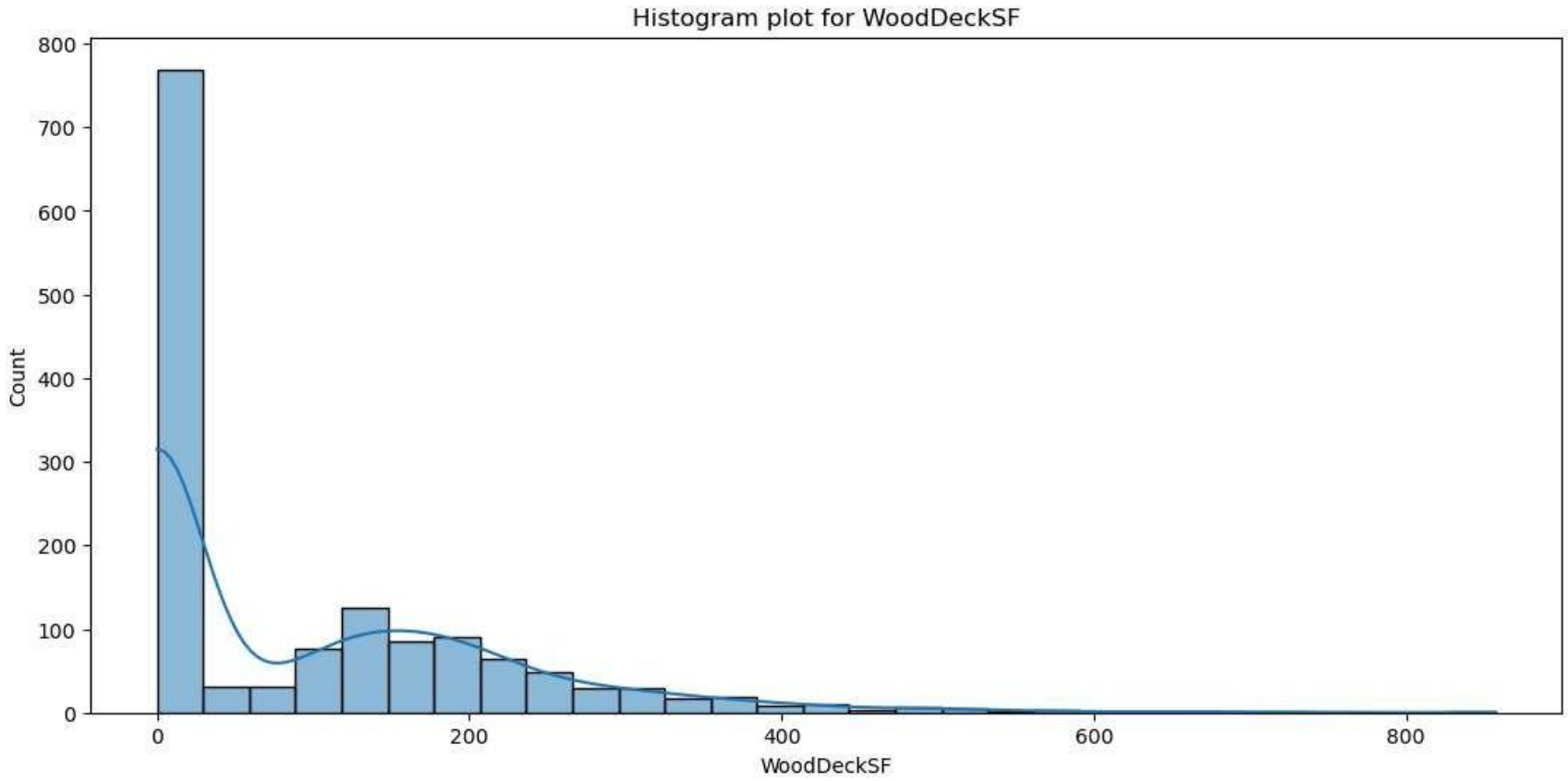

Histogram plot for TotRmsAbvGrd
Count
TotRmsAbvGrd
0
50
100
150
200
250
300
350
400
2
4
6
8
10
12
14

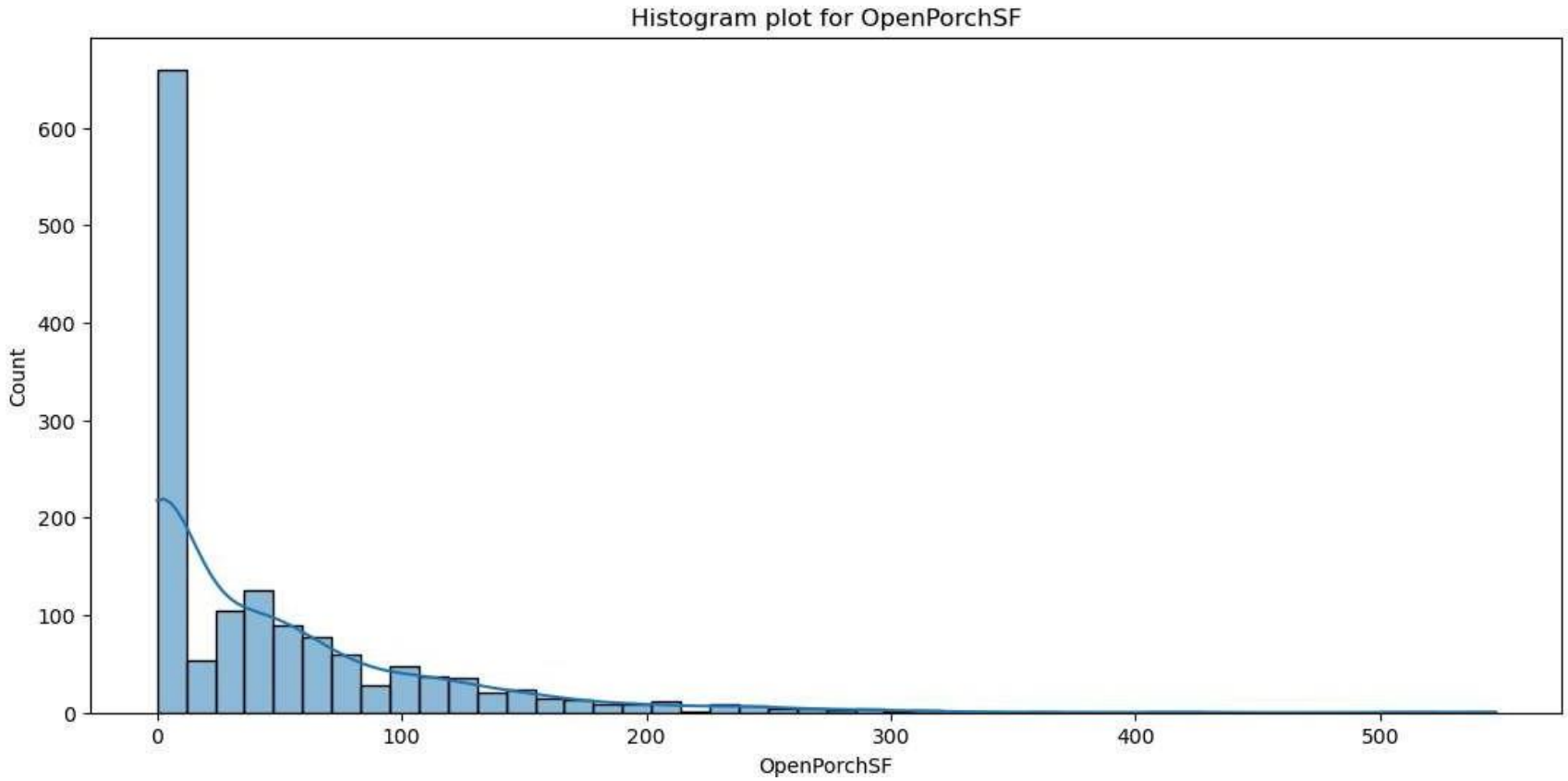

Histogram plot for Fireplaces
Count
Fireplaces
0
100
200
300
400
500
600
700
0.0
0.5
1.0
1.5
2.0
2.5
3.0

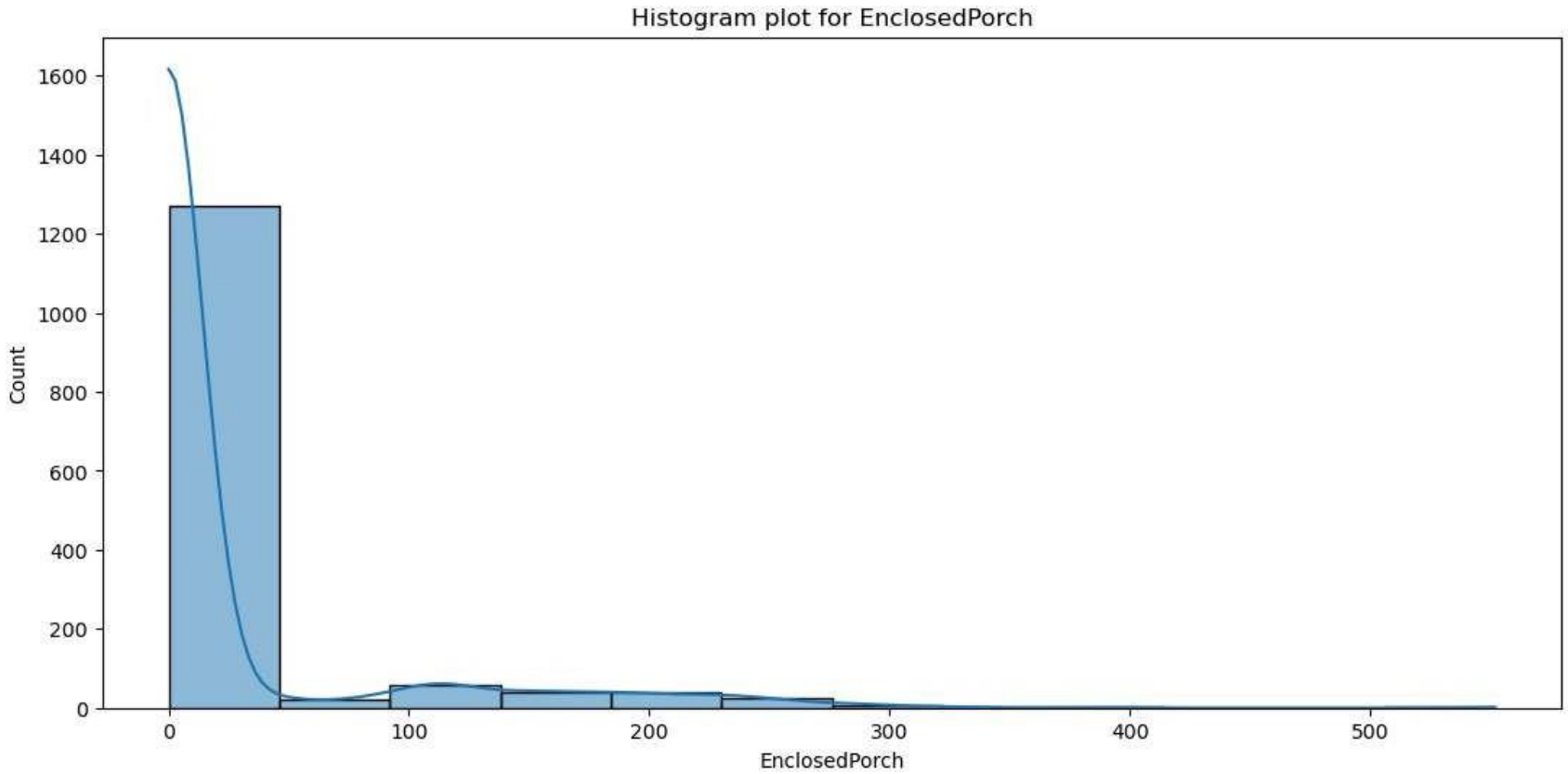

Histogram plot for GarageYrBlt
Count
GarageYrBlt
0
50
100
150
200
250
300
1900
1920
1940
1960
1980
2000

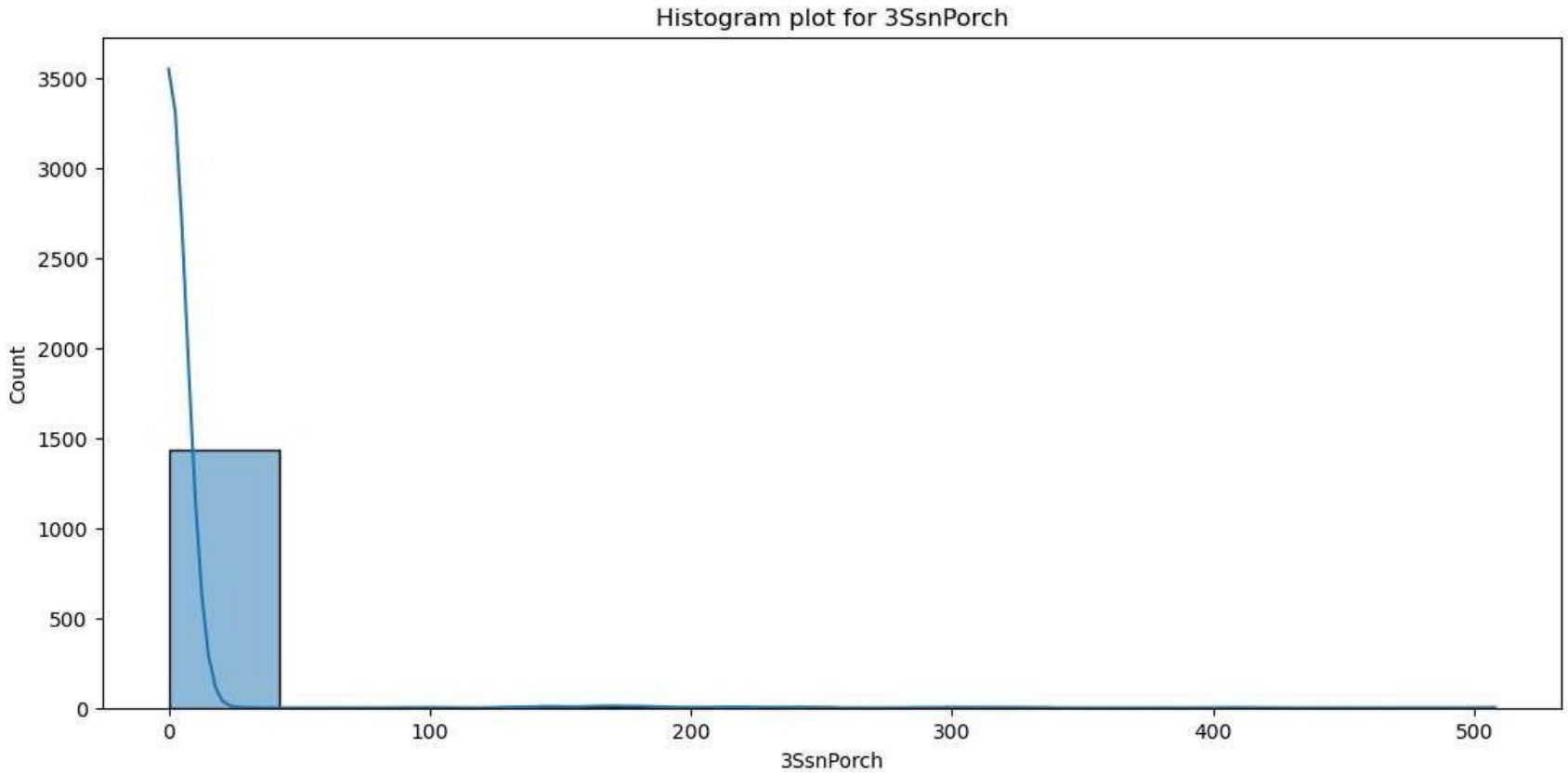

Histogram plot for GarageCars
Count
GarageCars
0
100
200
300
400
500
600
700
800
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

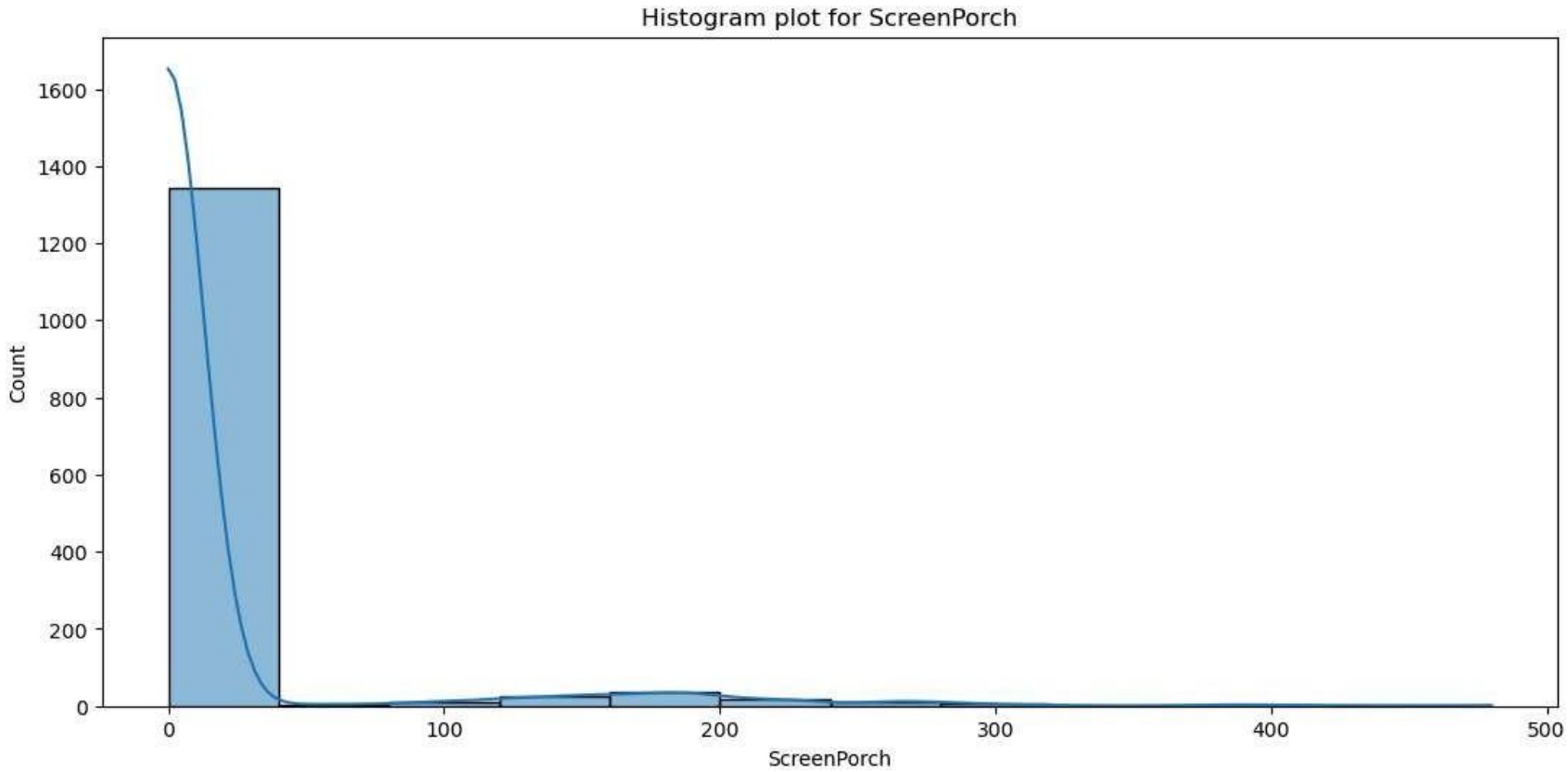

Histogram plot for GarageArea
Count
GarageArea
0
20
40
60
80
100
120
140
160
0
200
400
600
800
1000
1200
1400

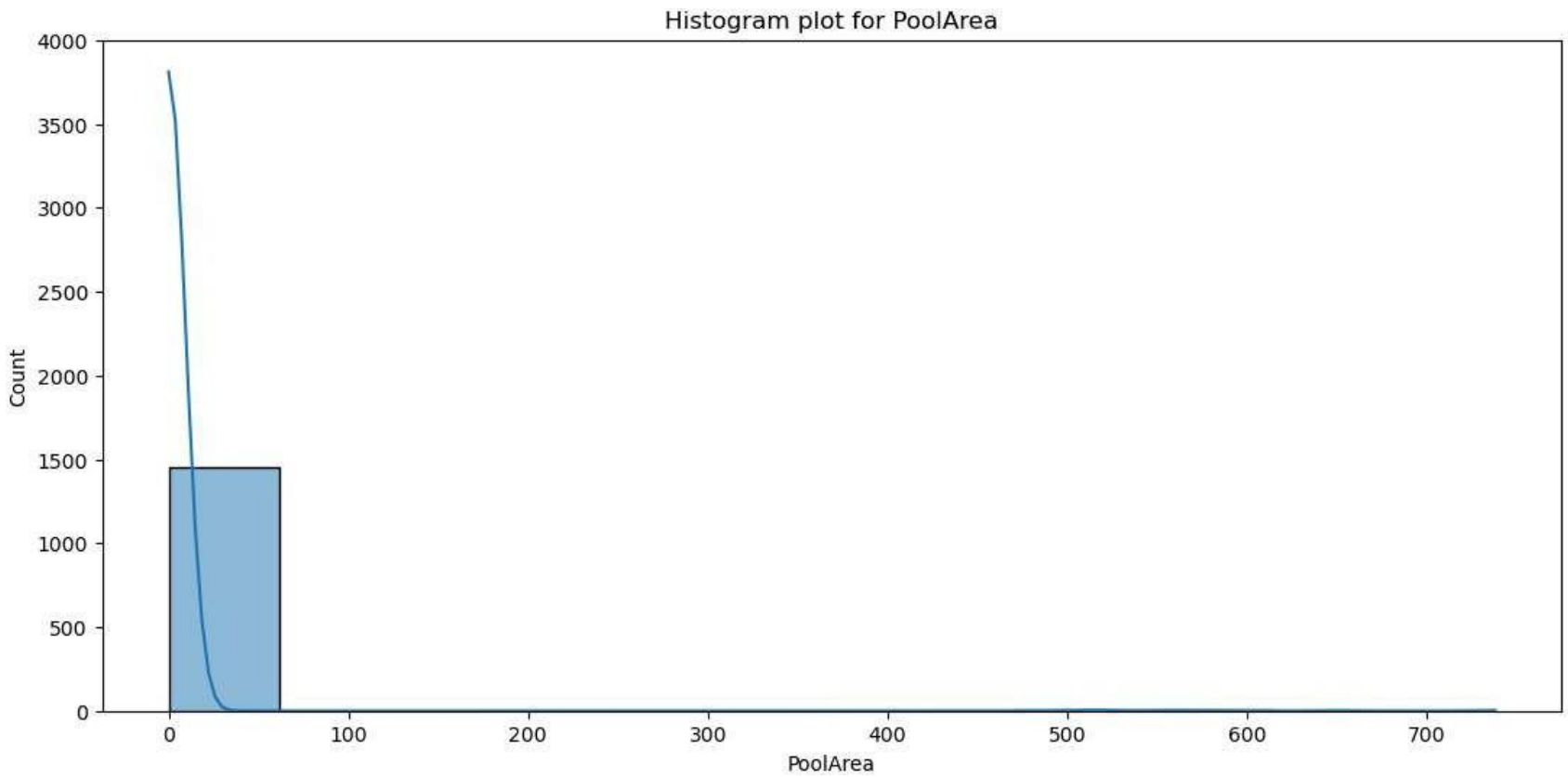

Histogram plot for WoodDeckSF
Count
WoodDeckSF
0
100
200
300
400
500
600
700
800
0
200
400
600
800

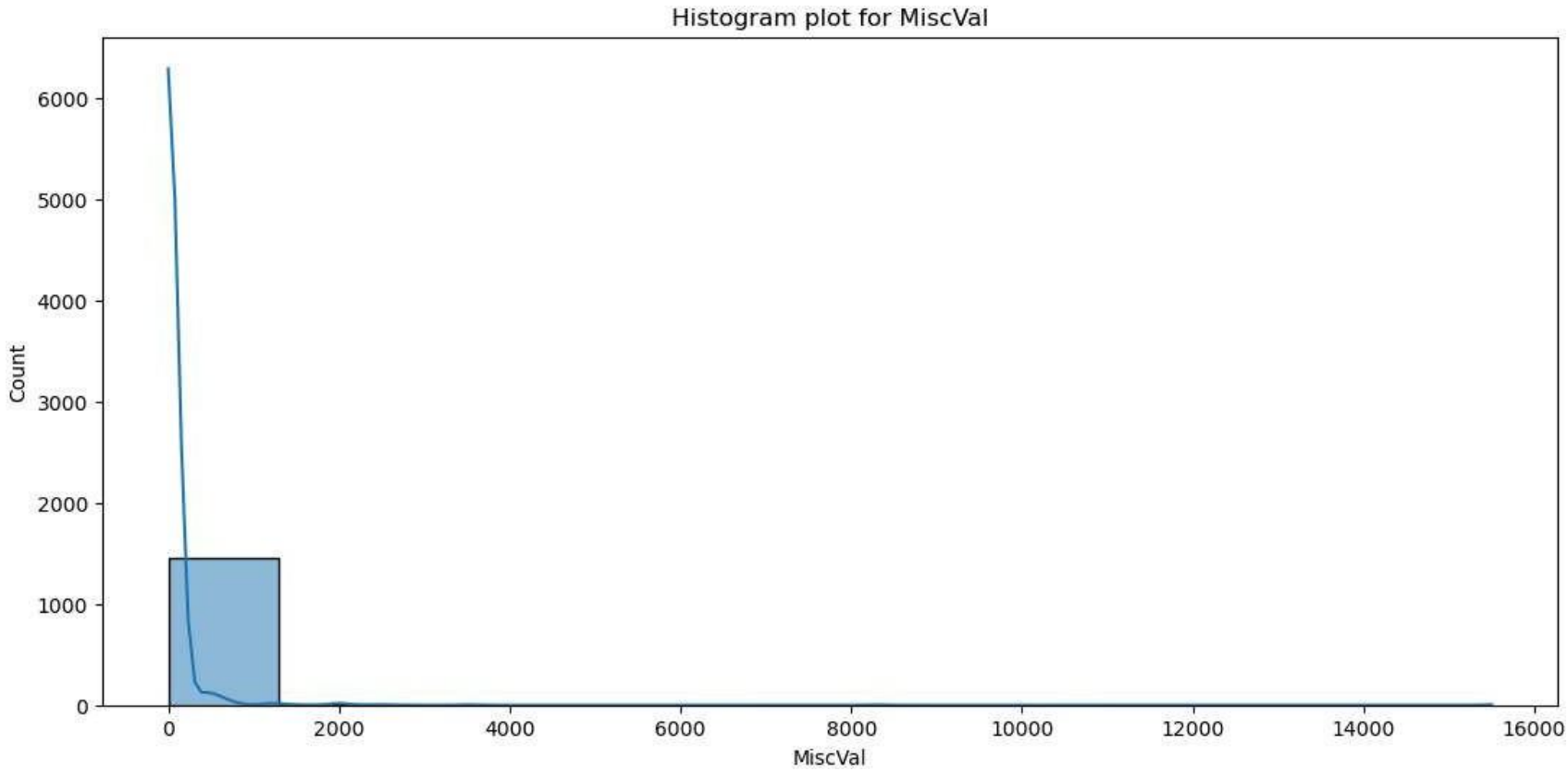

Histogram plot for OpenPorchSF
Count
OpenPorchSF
0
100
200
300
400
500
600

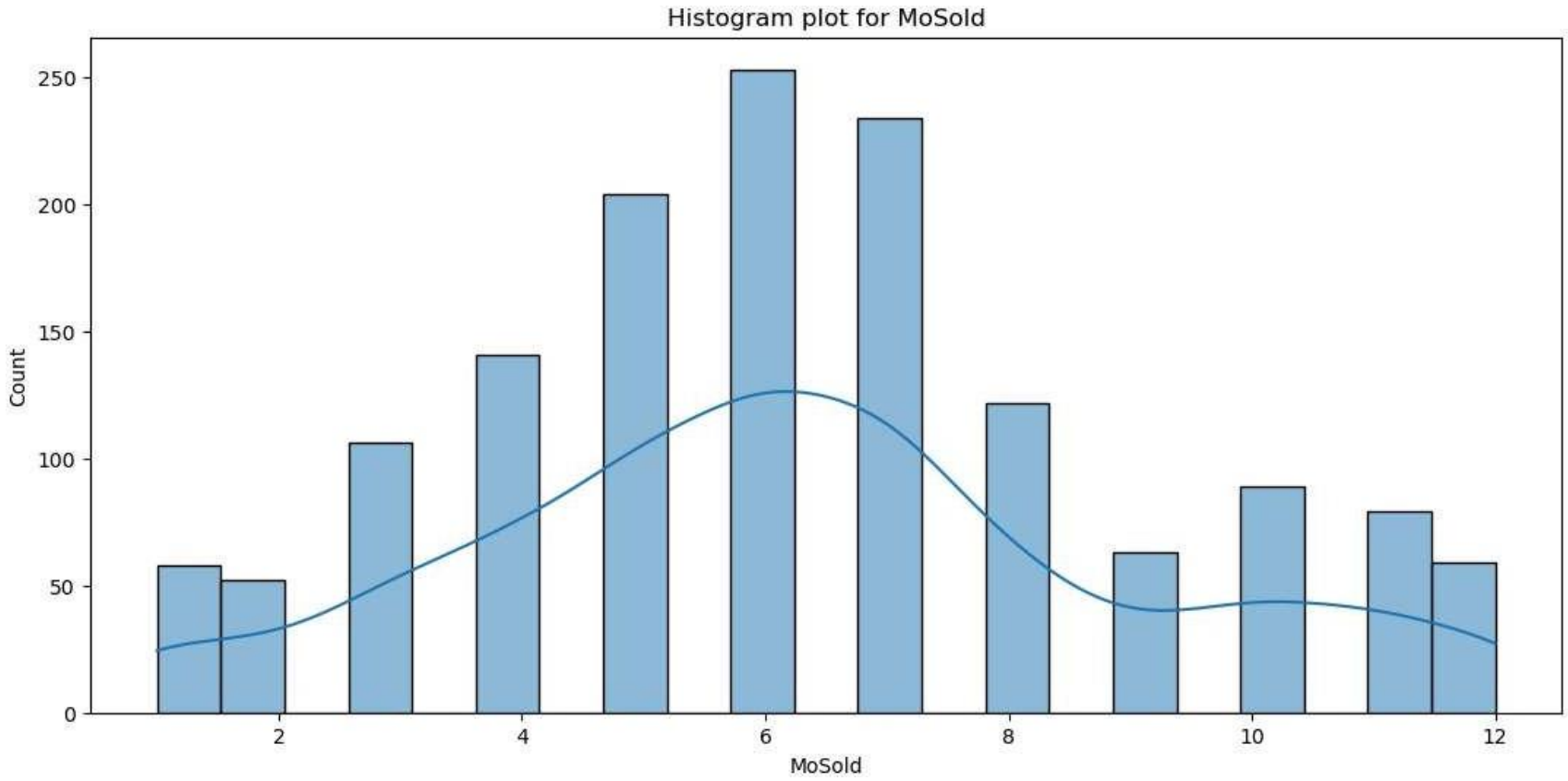

Histogram plot for EnclosedPorch
Count
EnclosedPorch
0
200
400
600
800
1000
1200
1400
1600
0
100
200
300
400
500

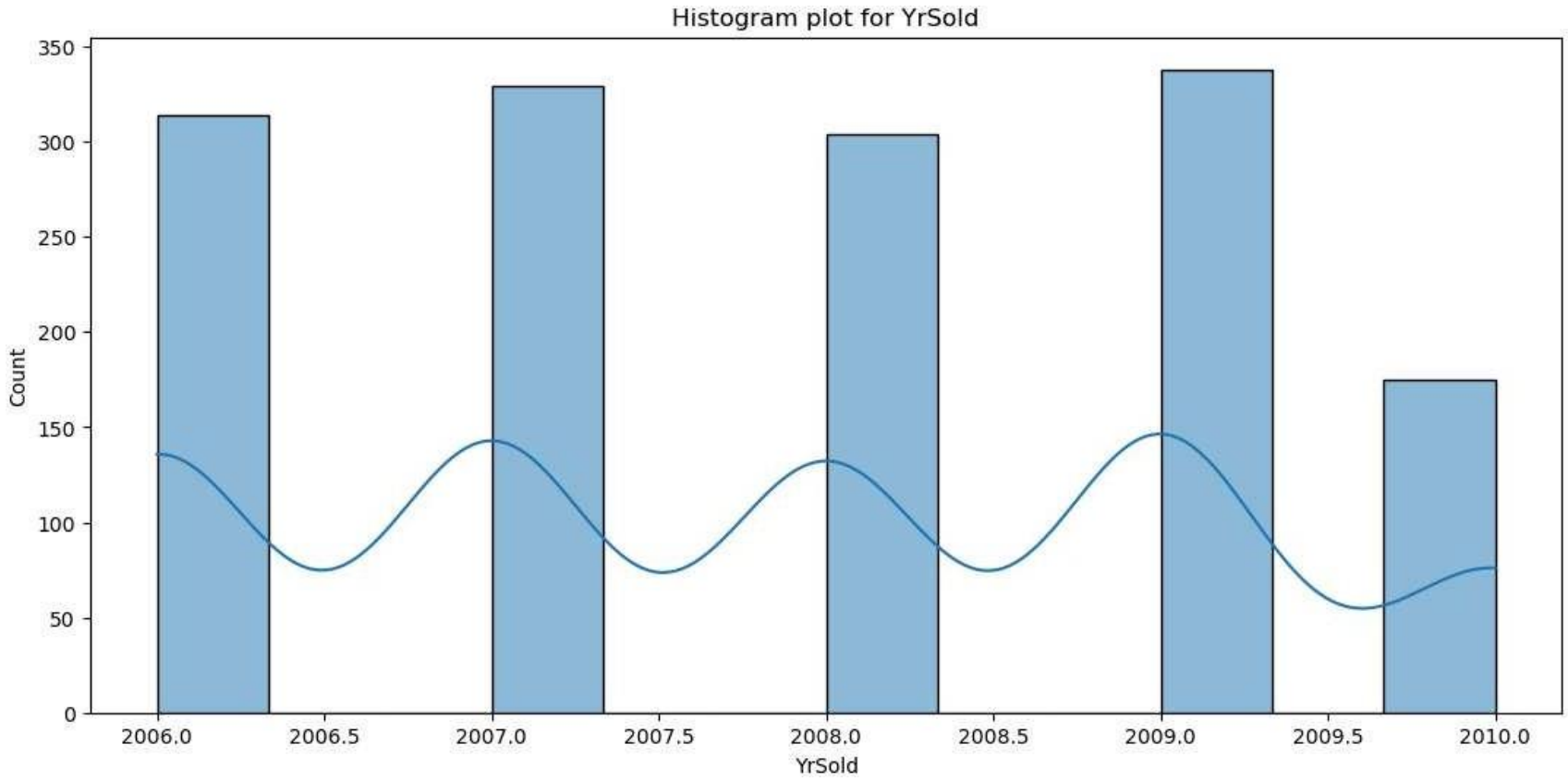

Histogram plot for 3SsnPorch
Count
3SsnPorch
0
500
1000
1500
2000
2500
3000
3500
0
100
200
300
400
500

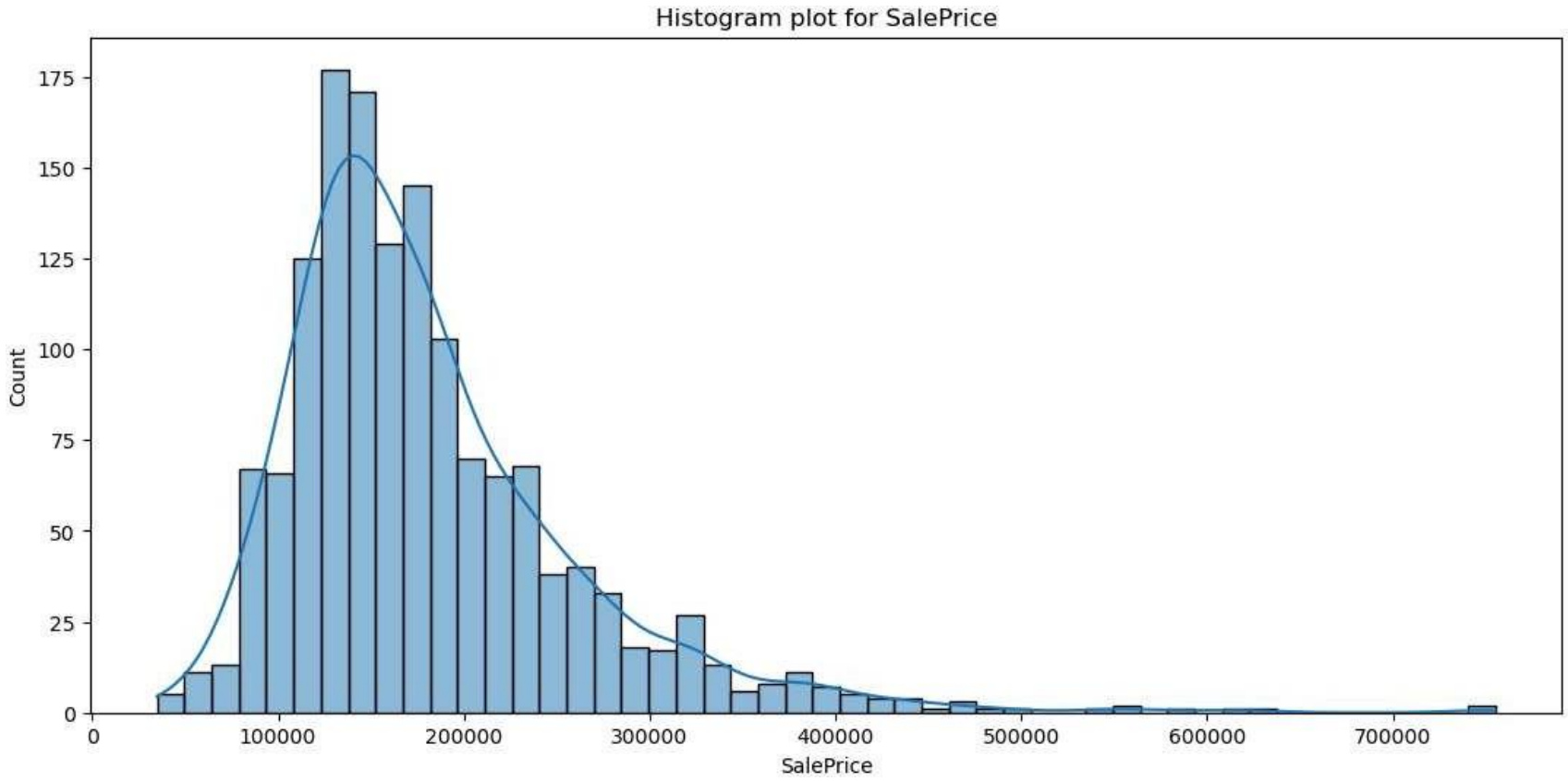

Histogram plot for ScreenPorch
Count
ScreenPorch
0
200
400
600
800
1000
1200
1400
1600
0
100
200
300
400
500


Histogram plot for PoolArea
Count
PoolArea
0
500
1000
1500
2000
2500
3000
3500
4000
0
100
200
300
400
500
600
700


Histogram plot for MiscVal
Count
MiscVal
0
1000
2000
3000
4000
5000
6000
0
2000
4000
6000
8000
10000
12000
14000
16000


Histogram plot for MoSold
Count
MoSold
0
50
100
150
200
250
2
4
6
8
10
12


Histogram plot for YrSold
Count
YrSold
0
50
100
150
200
250
300
350
2006.0
2006.5
2007.0
2007.5
2008.0
2008.5
2009.0
2009.5
2010.0

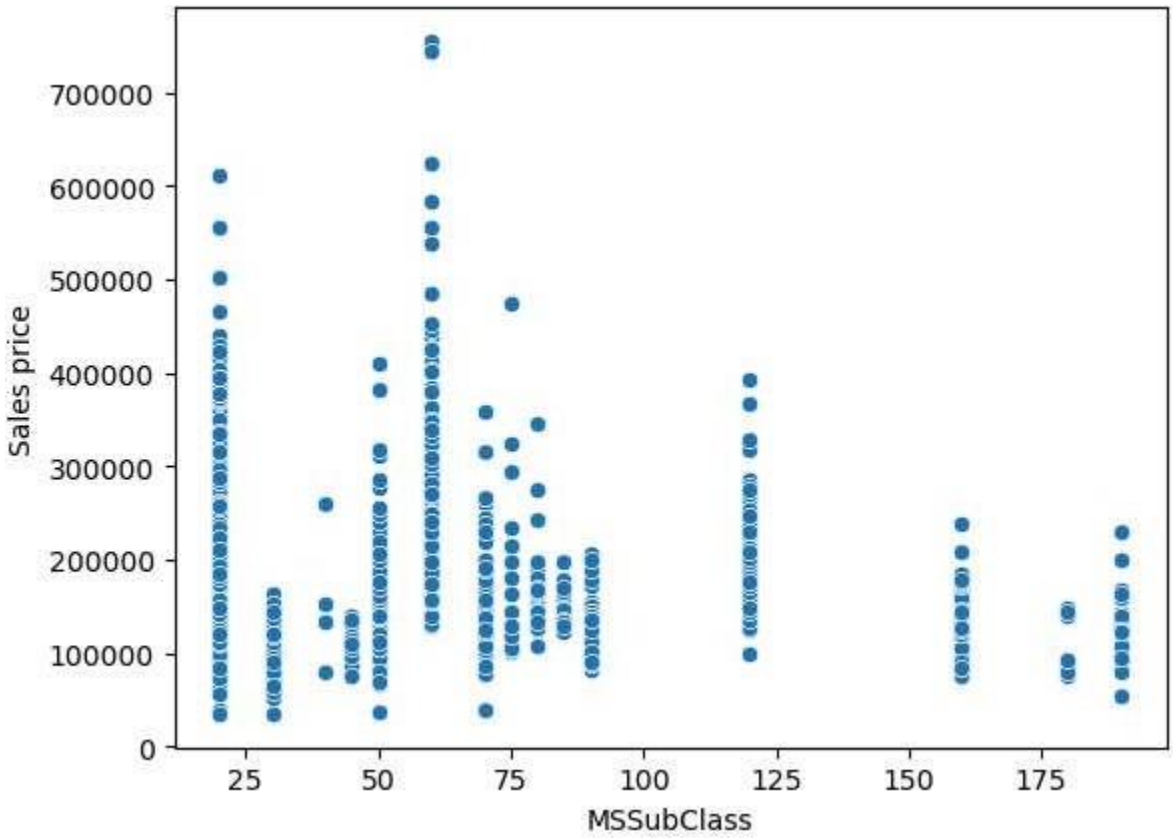


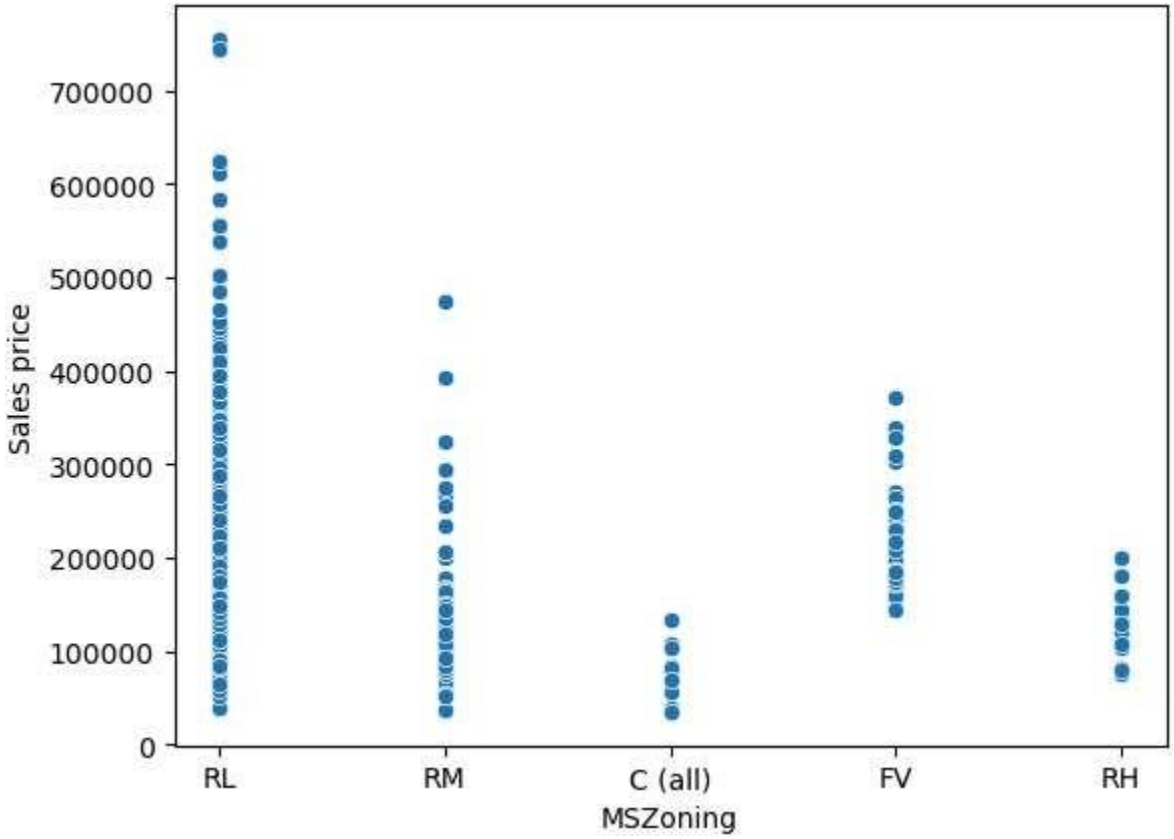
Bivariate Analysis

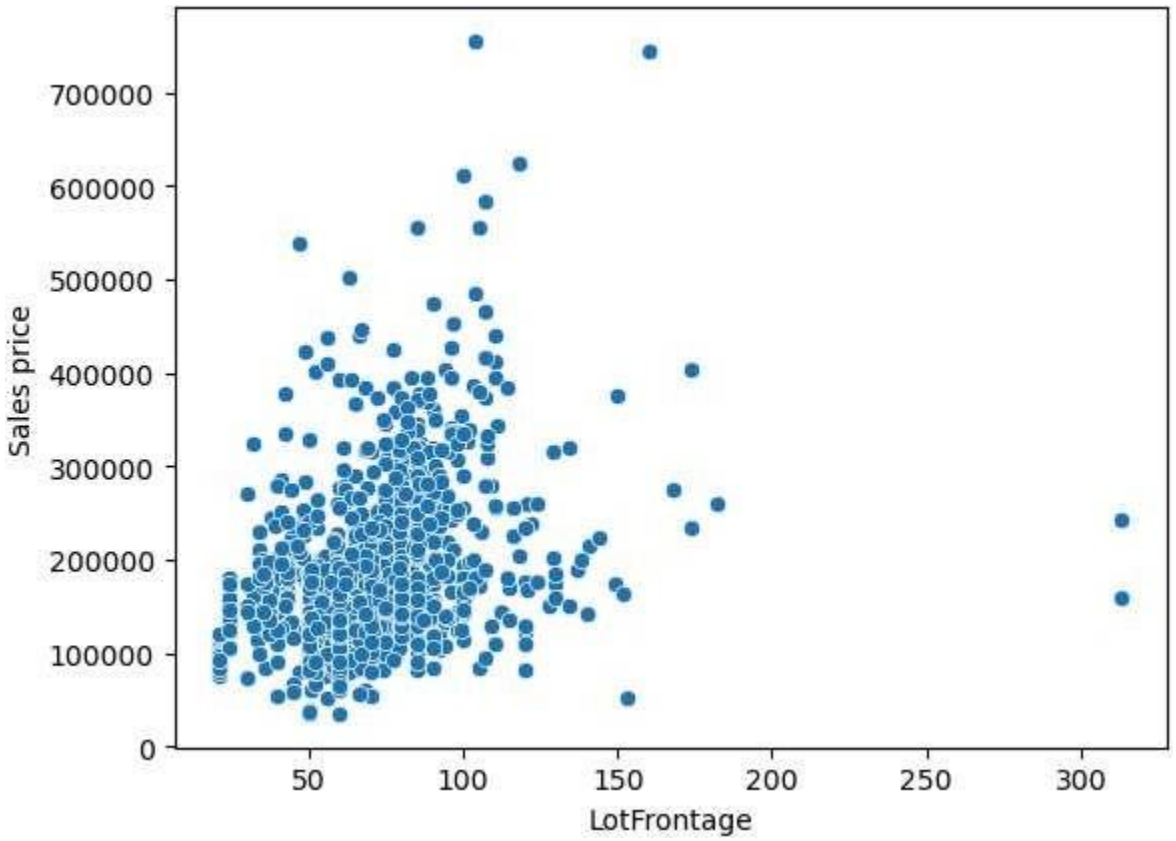
In [11]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

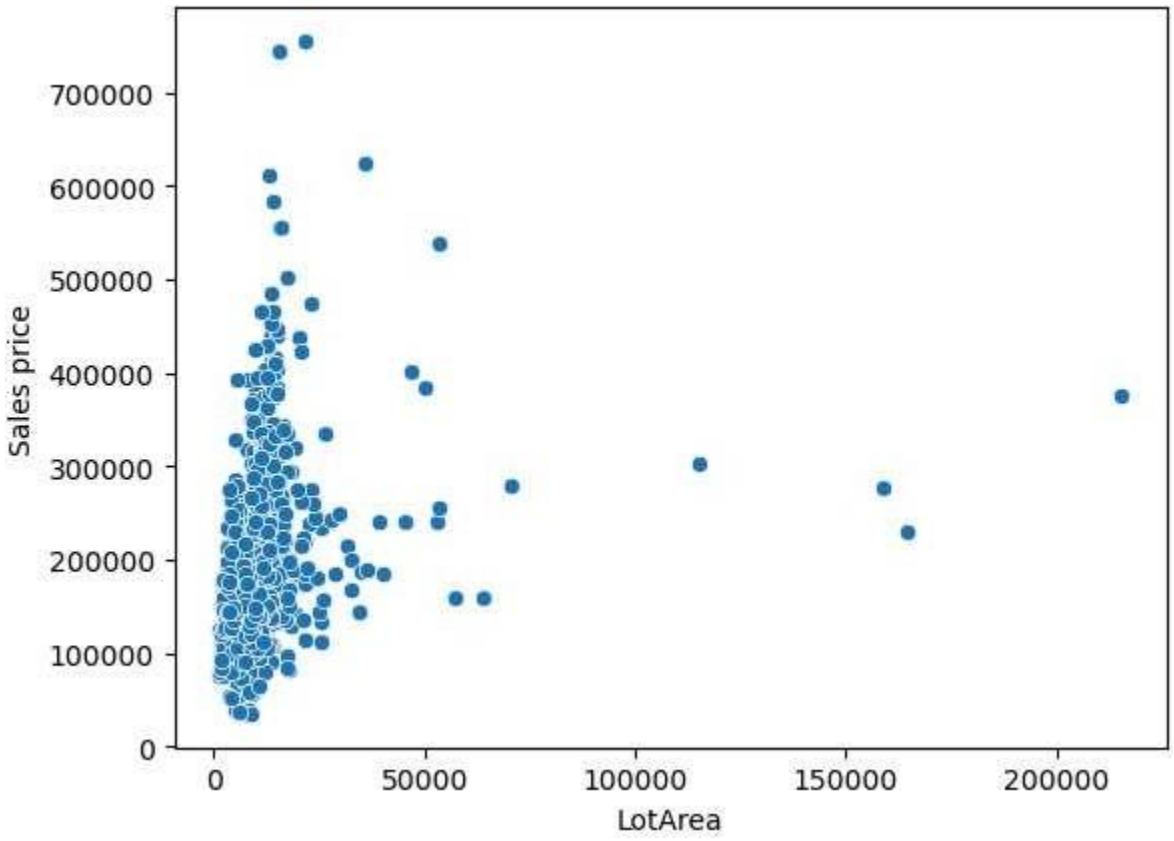
contineous vs contineous

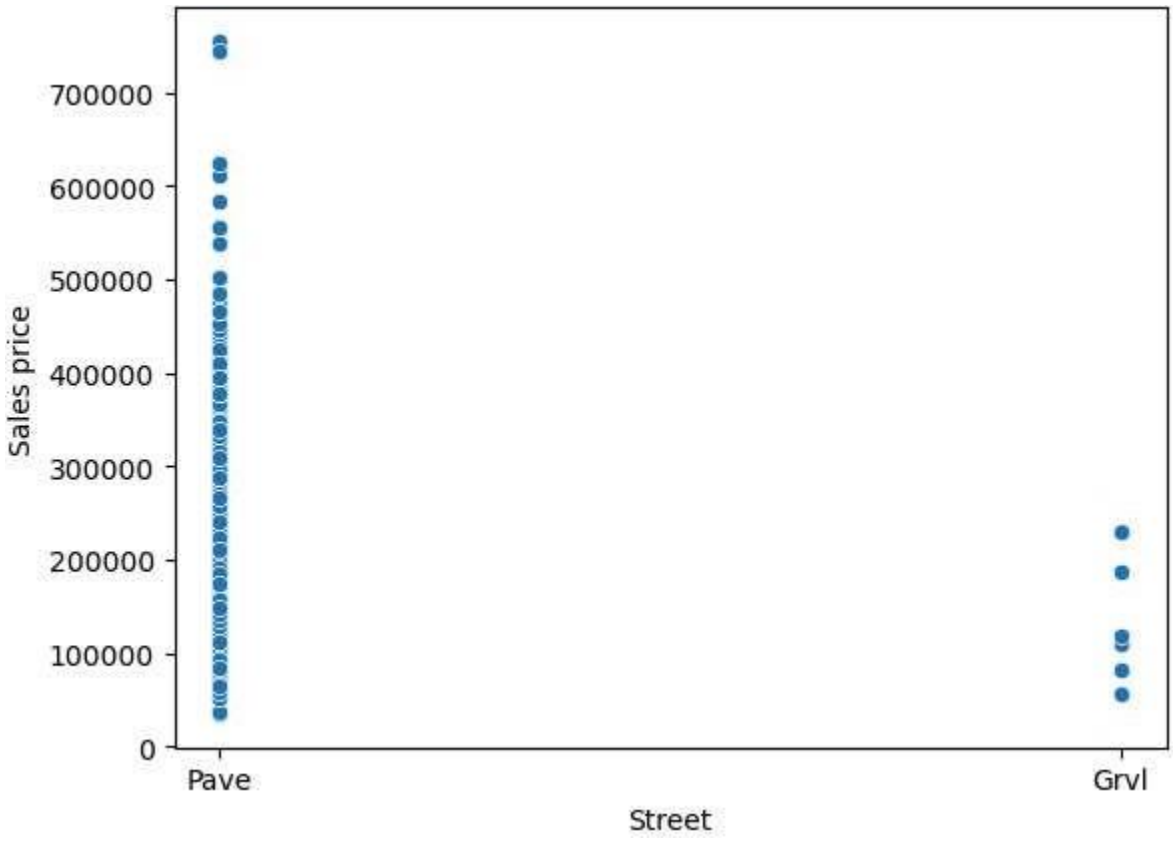
In [12]:
```
for i in df.drop(columns=['Id','SalePrice']):
    sns.scatterplot(data=df, x=i, y='SalePrice')
    plt.xlabel(i)
    plt.ylabel('Sales price')
    plt.show()
```

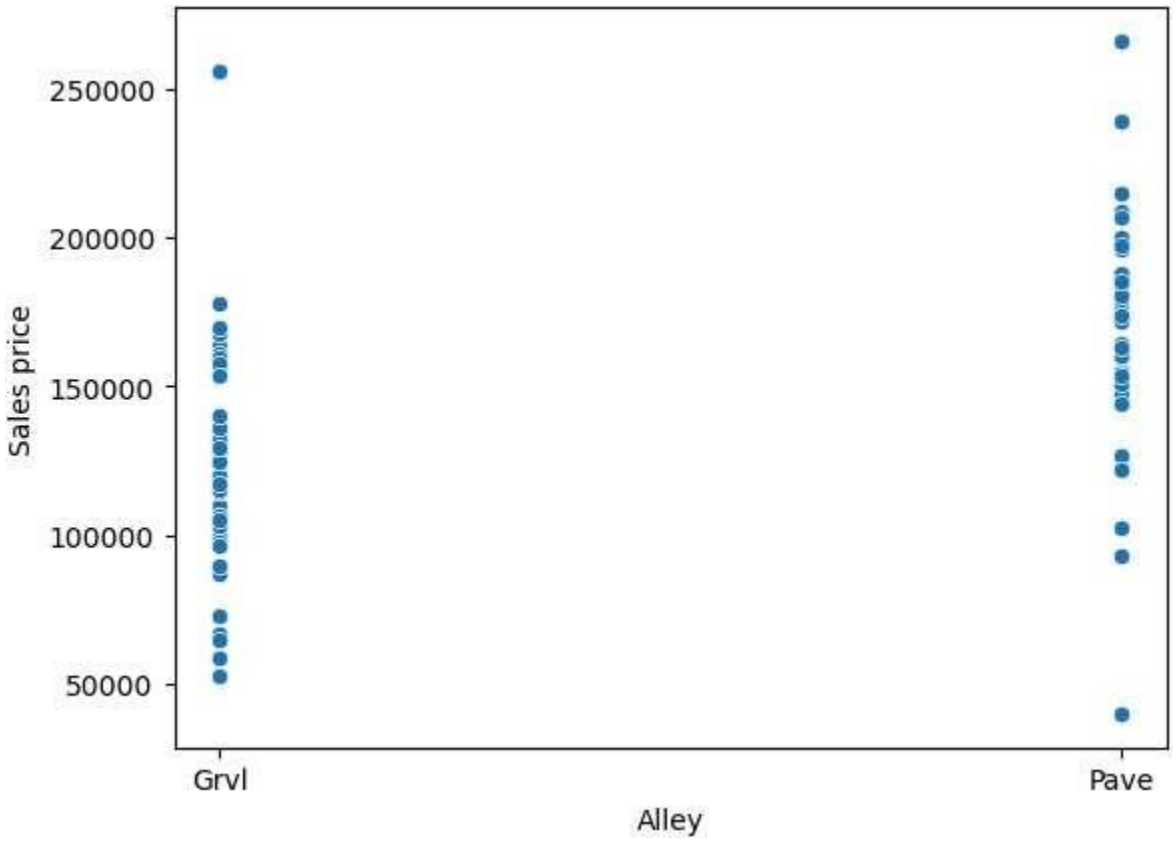

Sales price
MSSubClass
0
100000
200000
300000
400000
500000
600000
700000
25
50
75
100
125
150
175

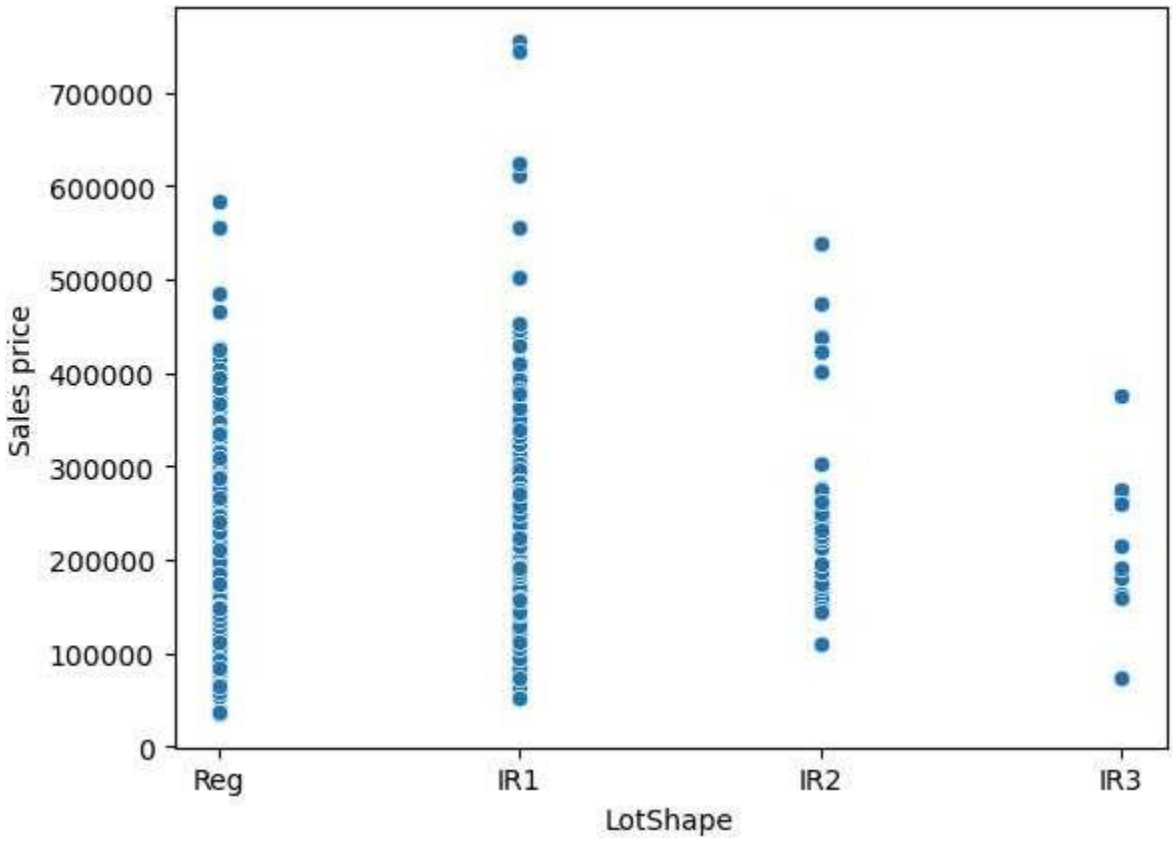

Sales price
700000
600000
500000
400000
300000
200000
100000
0
RL
RM
C (all)
FV
RH
MSZoning

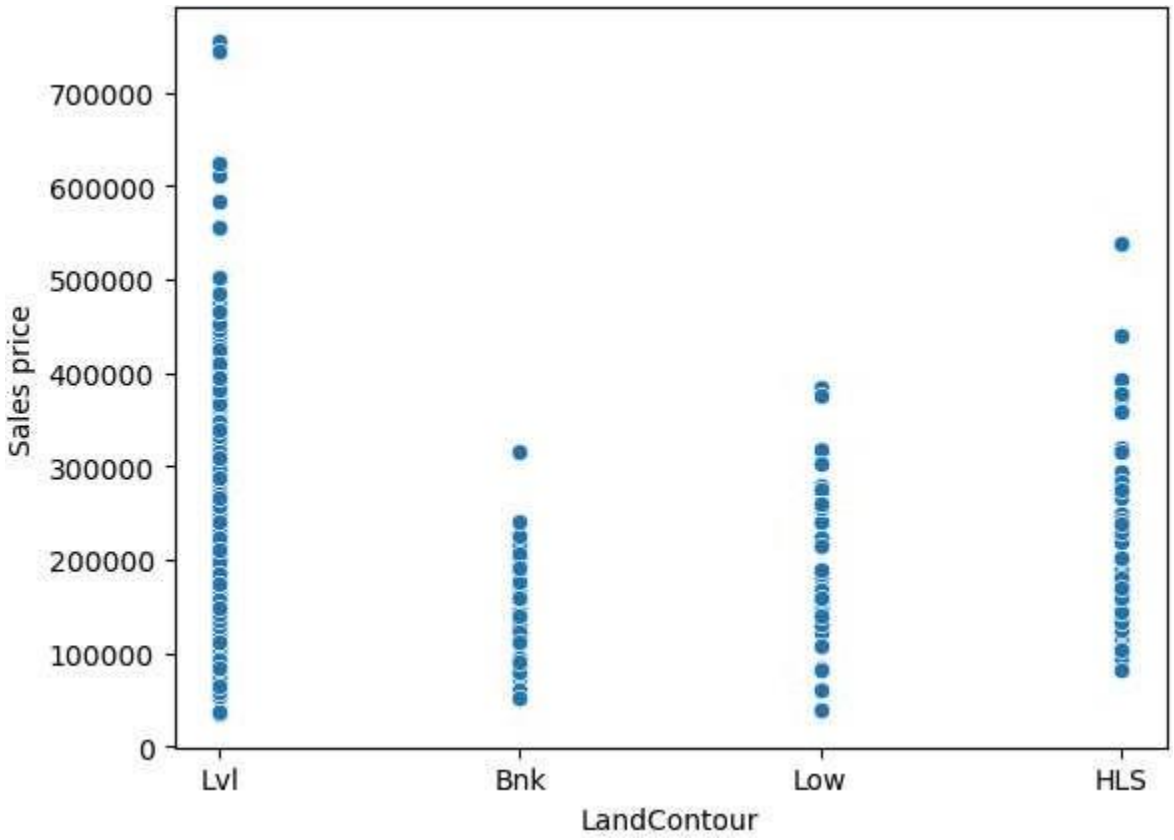

Sales price
700000
600000
500000
400000
300000
200000
100000
0
50
100
150
200
250
300
LotFrontage

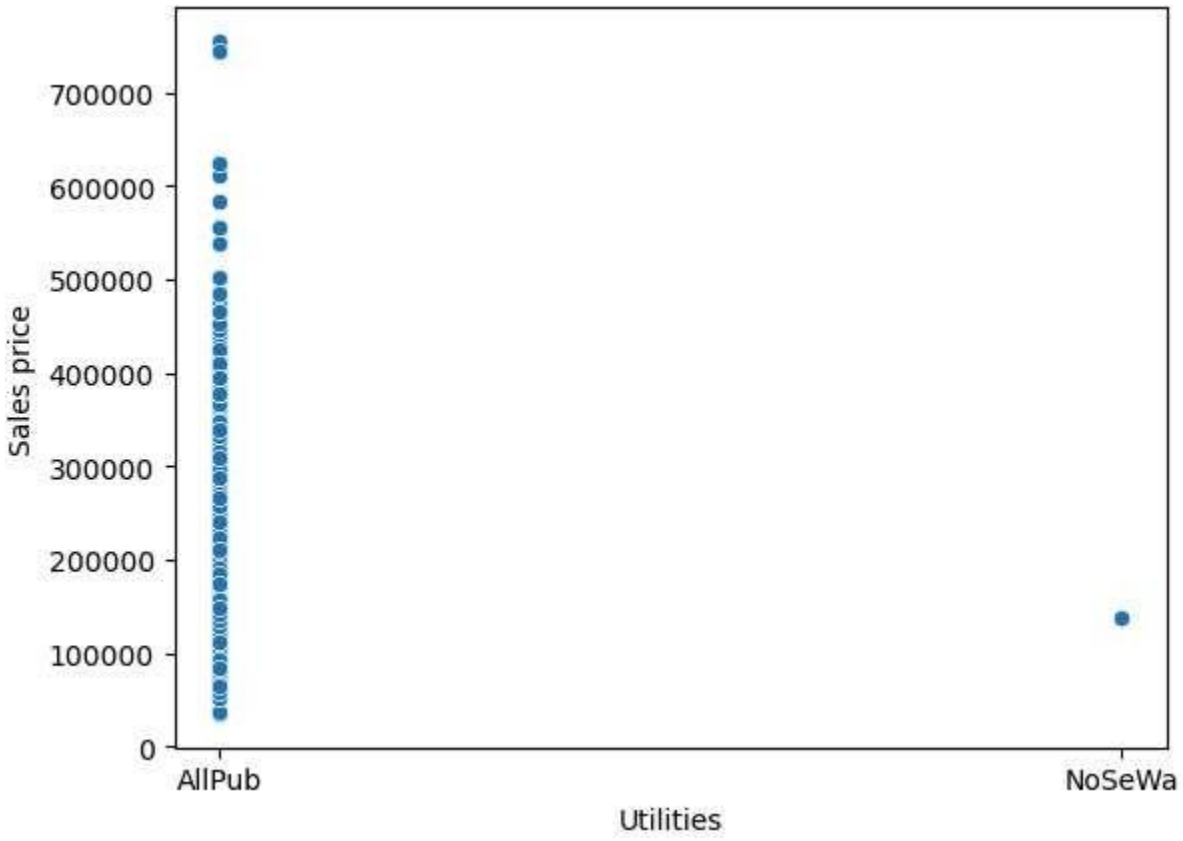

Sales price
700000
600000
500000
400000
300000
200000
100000
0
0
50000
100000
150000
200000
LotArea

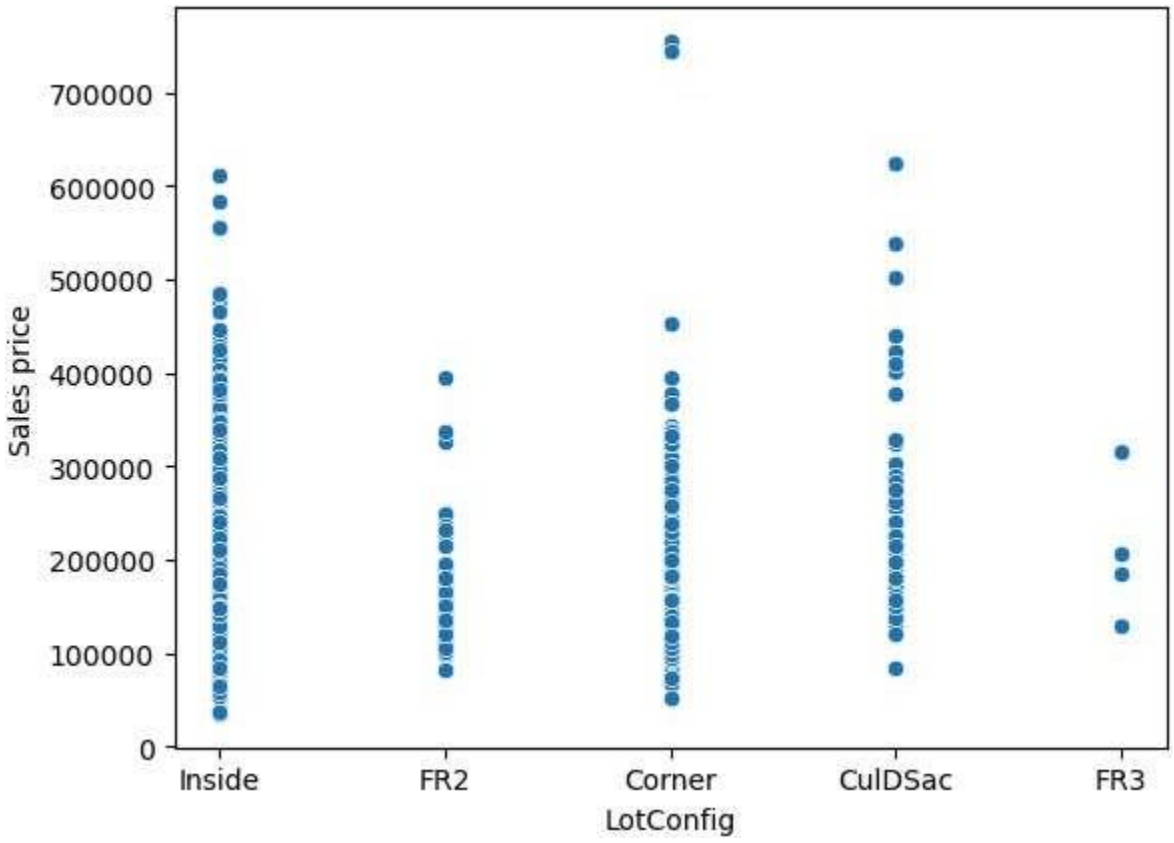

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Pave
Grvl
Street

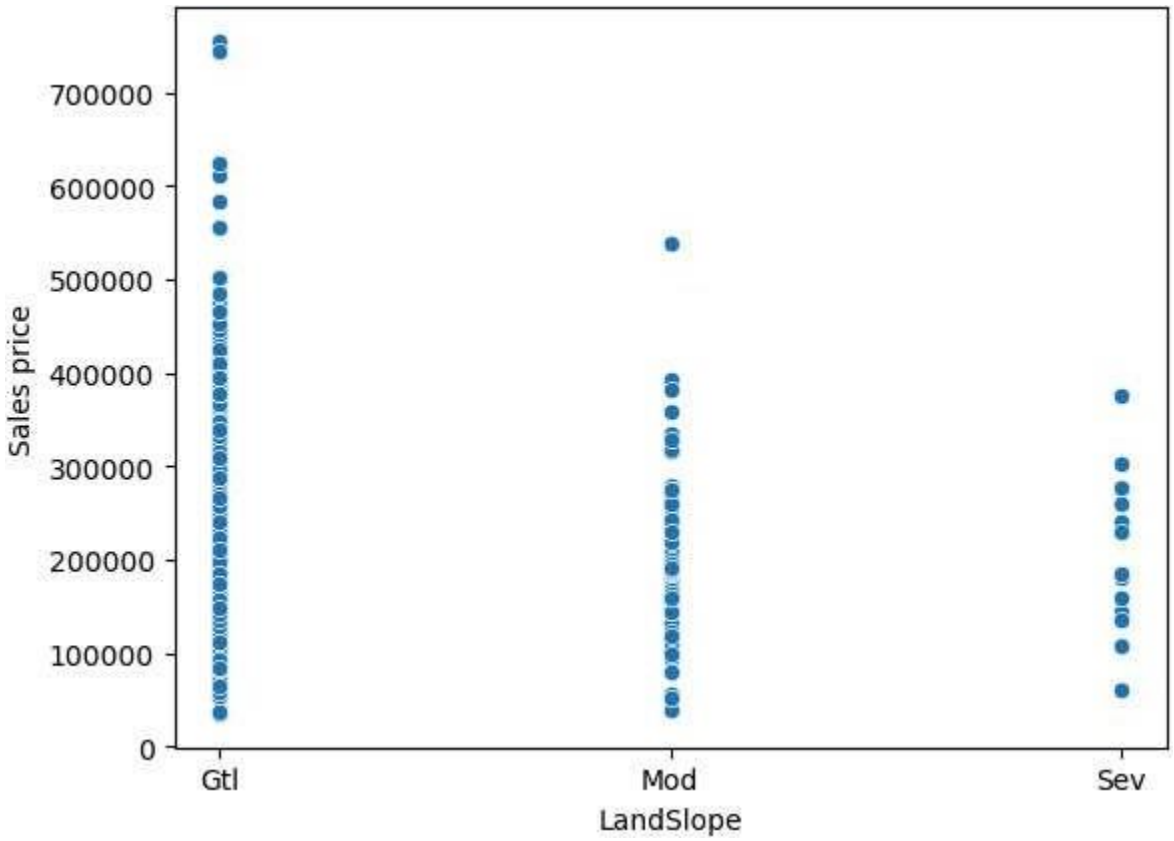

250000
200000
150000
100000
50000
Sales price
Grvl
Pave
Alley

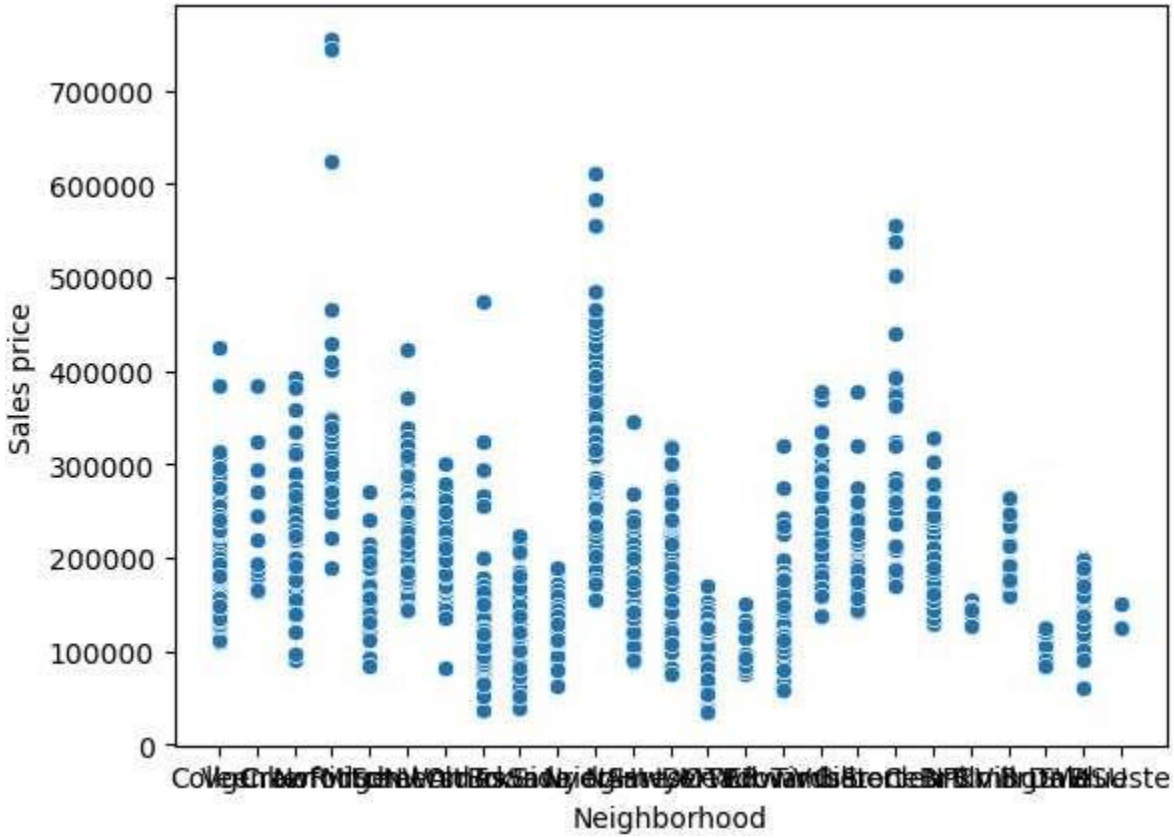

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Reg
IR1
IR2
IR3
LotShape

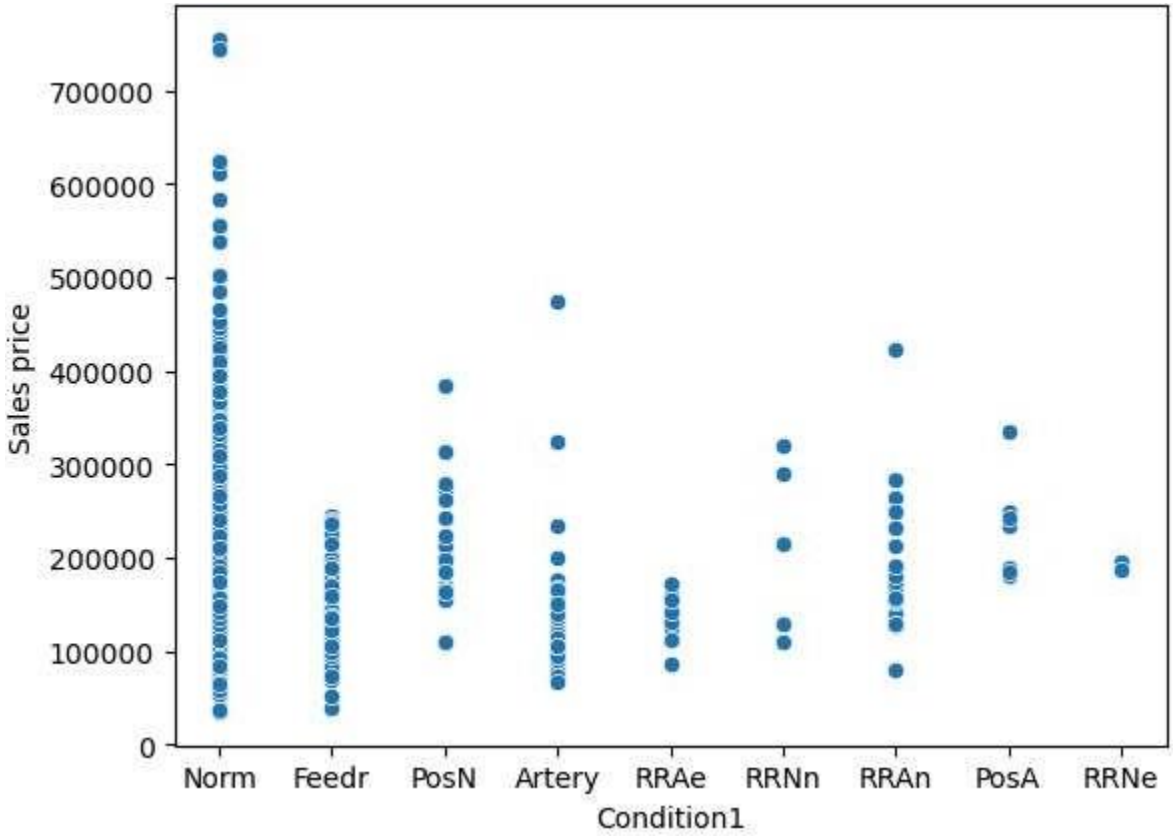

Sales price
700000
600000
500000
400000
300000
200000
100000
0
Lvl
Bnk
Low
HLS
LandContour

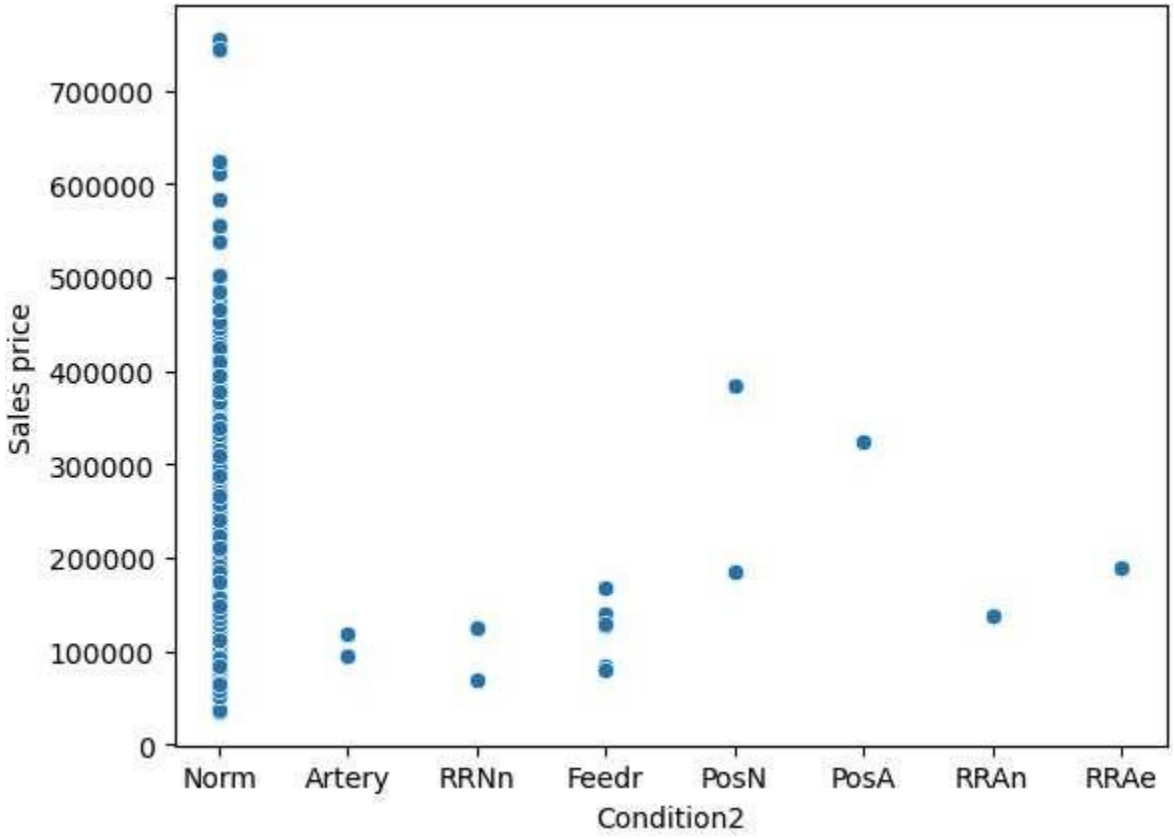

700000
600000
500000
400000
300000
200000
100000
0
Sales price
AllPub
NoSeWa
Utilities

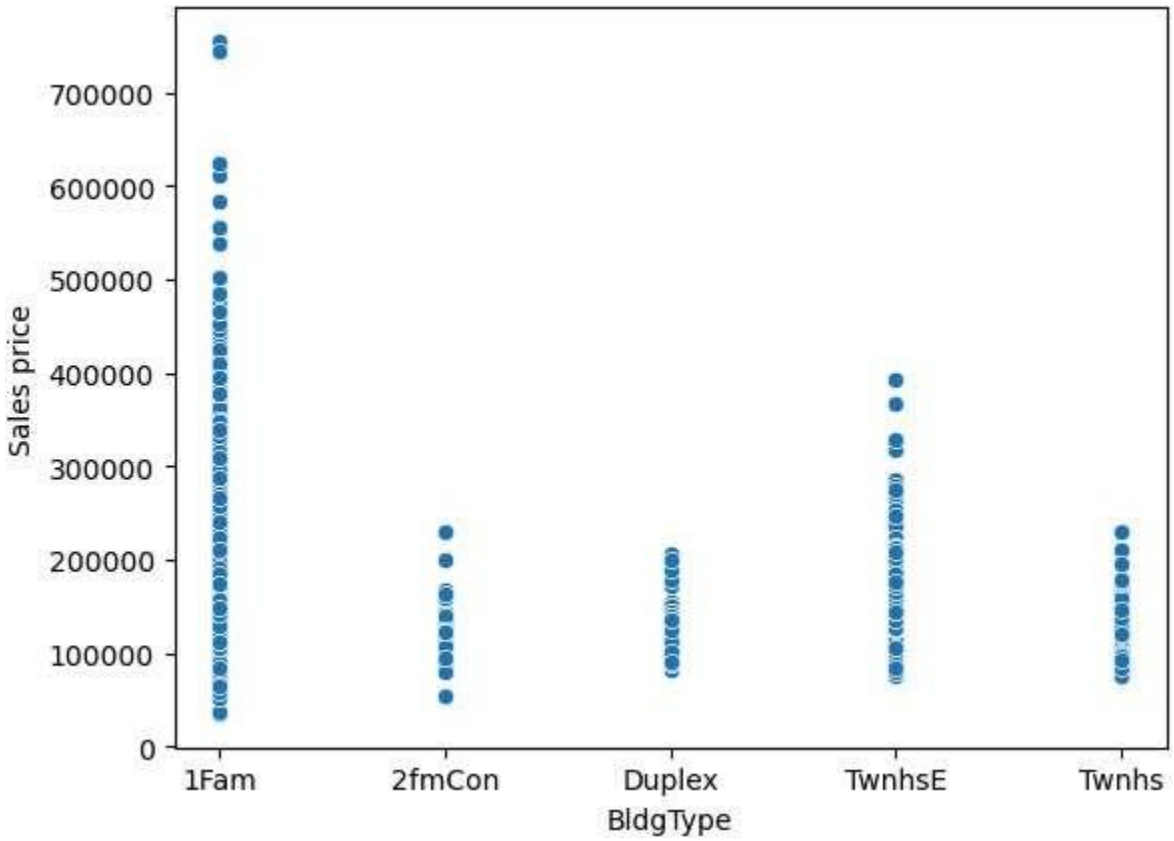

Sales price
700000
600000
500000
400000
300000
200000
100000
0
Inside
FR2
Corner
CulDSac
FR3
LotConfig

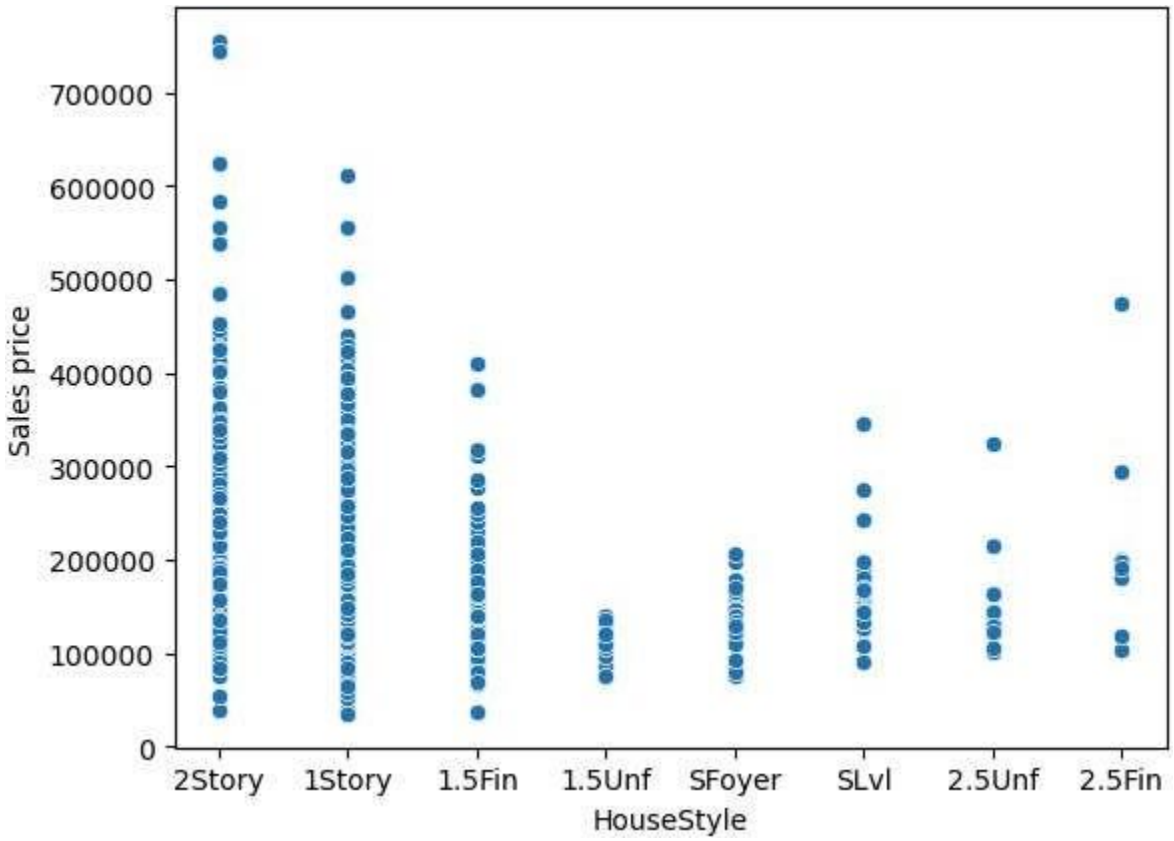

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Gtl
Mod
Sev
LandSlope

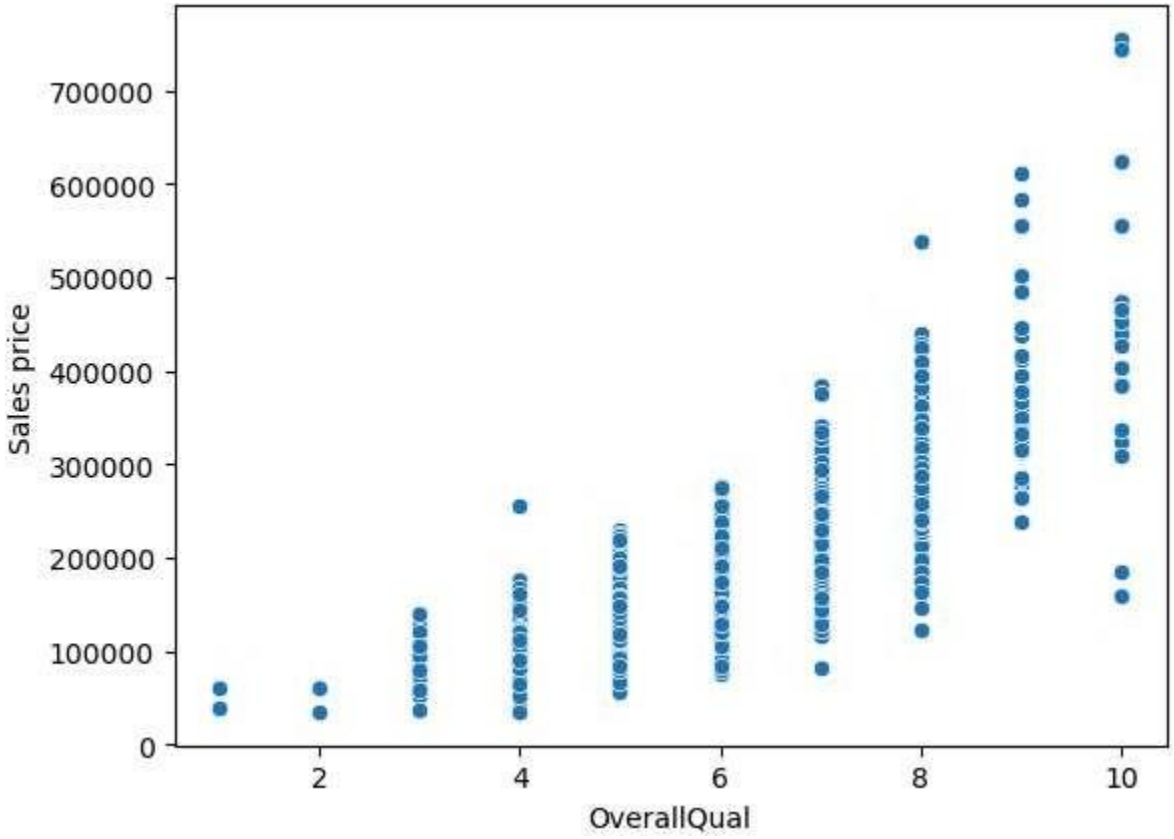

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Neighborhood

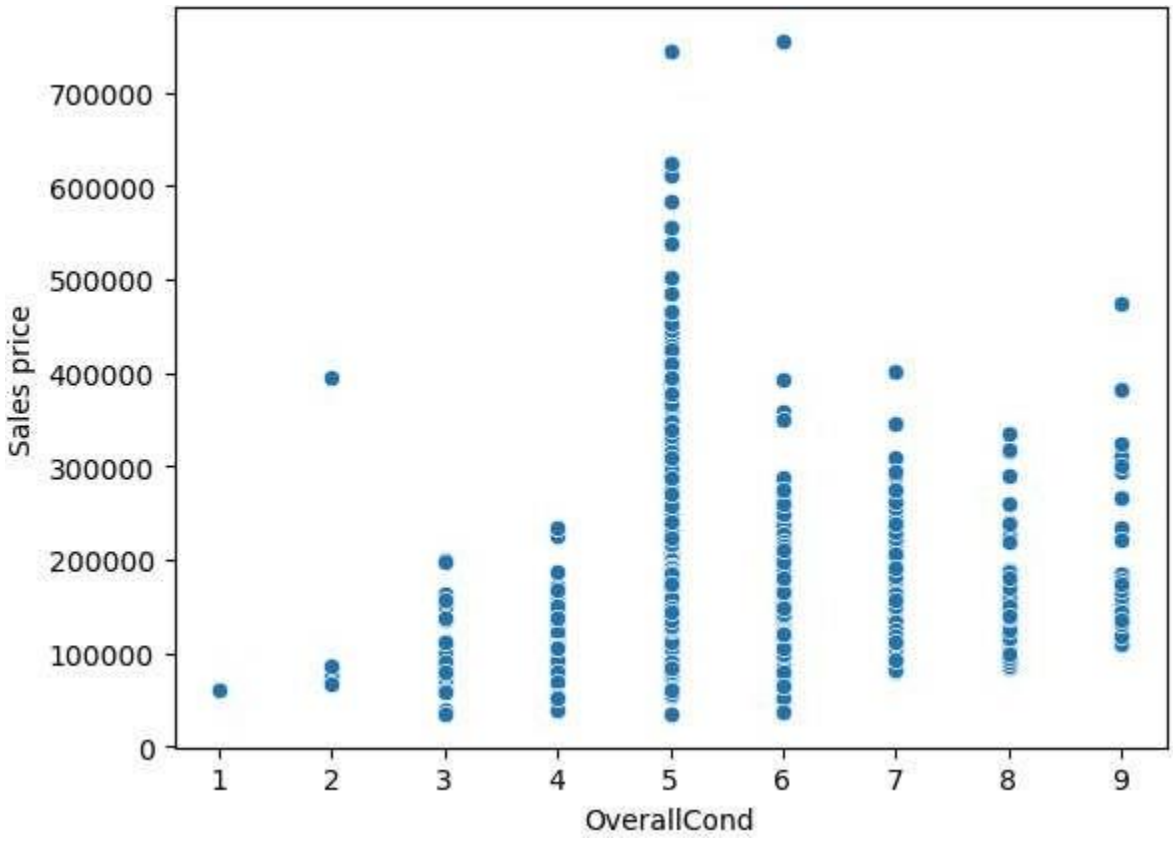

Sales price
700000
600000
500000
400000
300000
200000
100000
0
Norm
Feedr
PosN
Artery
RRAe
RRNn
RRAn
PosA
RRNe
Condition1

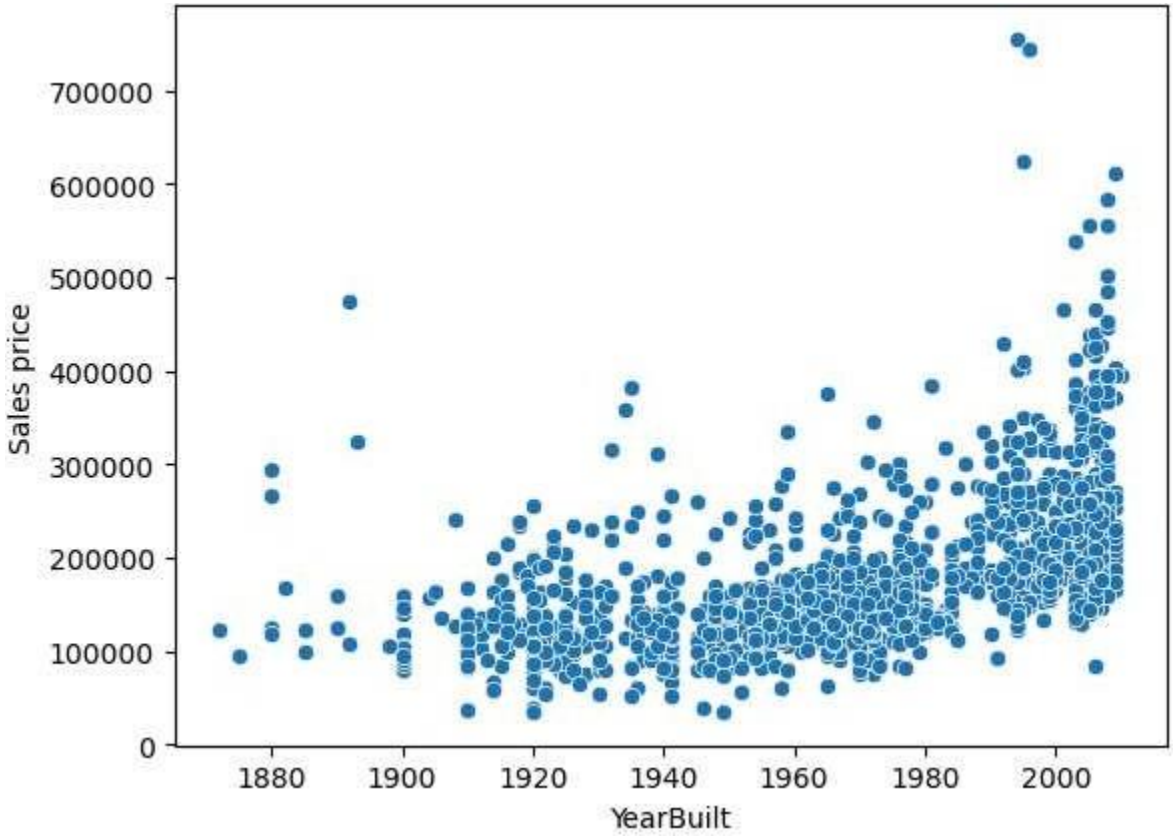

Sales price
700000
600000
500000
400000
300000
200000
100000
0
Norm
Artery
RRNn
Feedr
PosN
PosA
RRAn
RRAe
Condition2

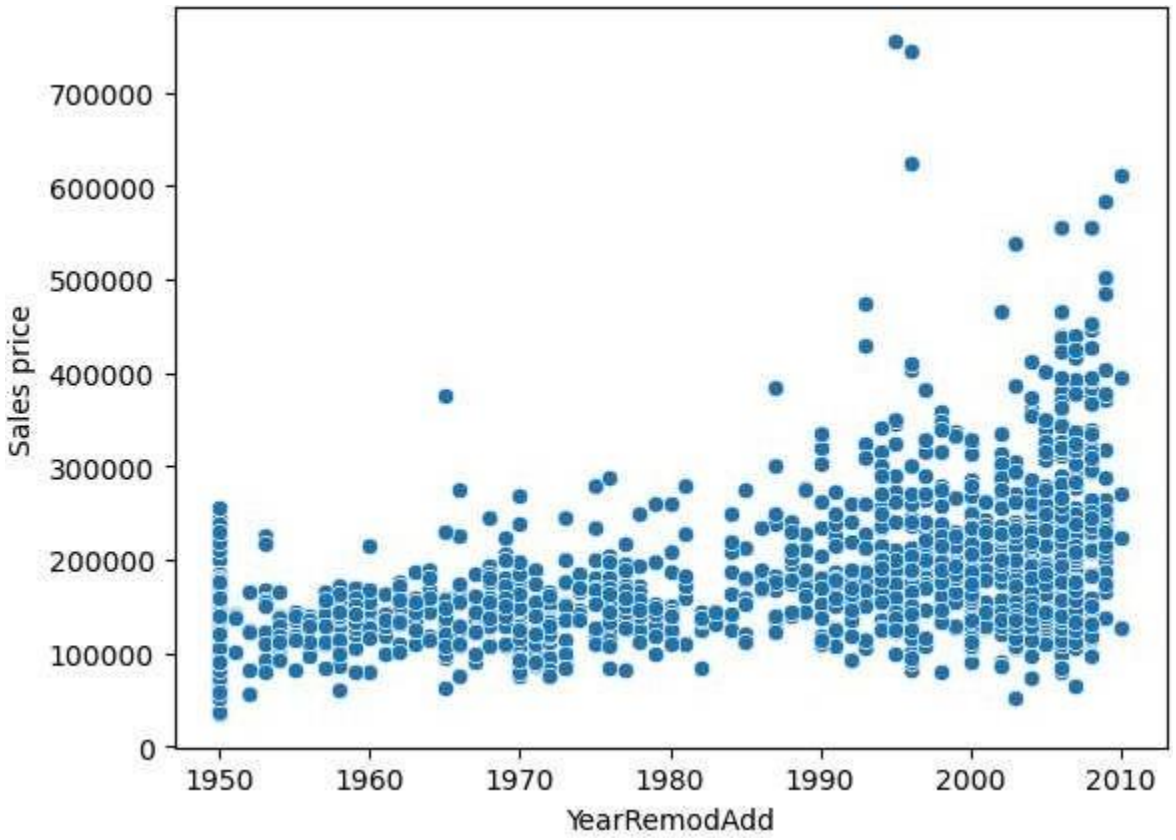

Sales price
700000
600000
500000
400000
300000
200000
100000
0
1Fam
2fmCon
Duplex
TwnhsE
Twnhs
BldgType

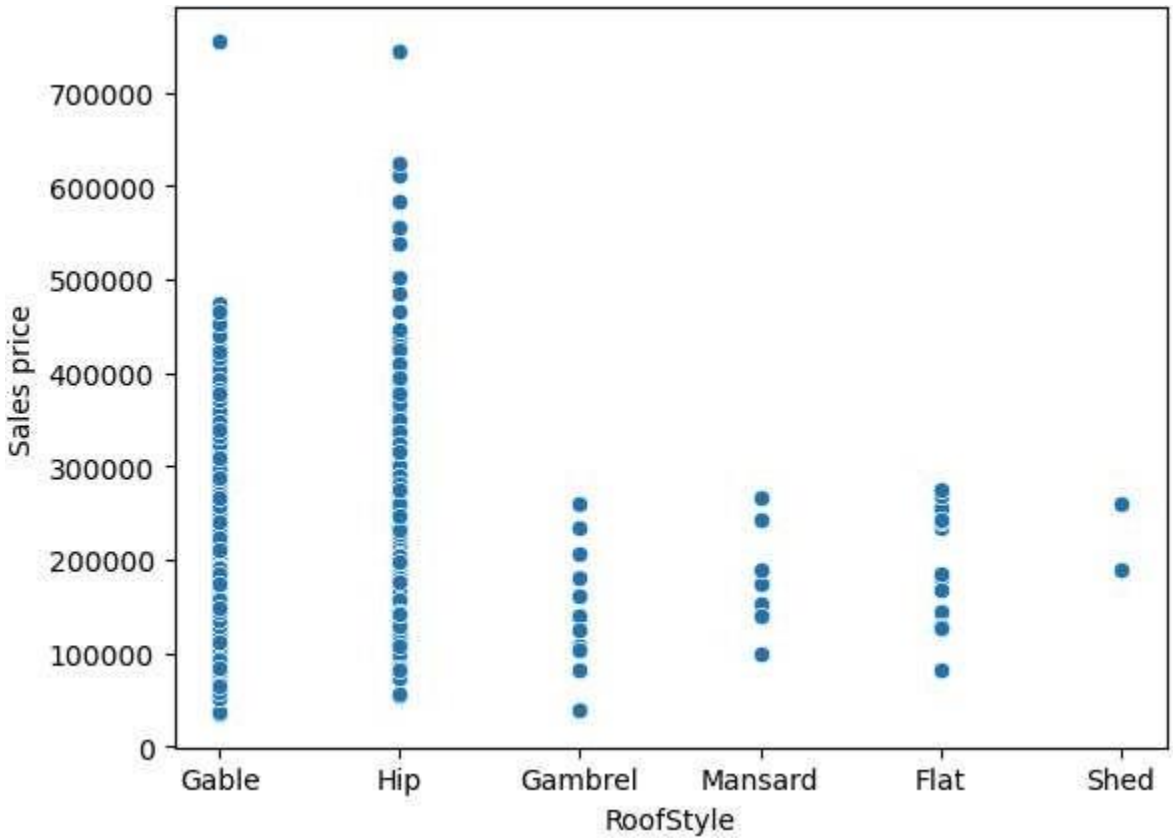

Sales price
700000
600000
500000
400000
300000
200000
100000
0
2Story
1Story
1.5Fin
1.5Unf
SFoyer
SLvl
2.5Unf
2.5Fin
HouseStyle

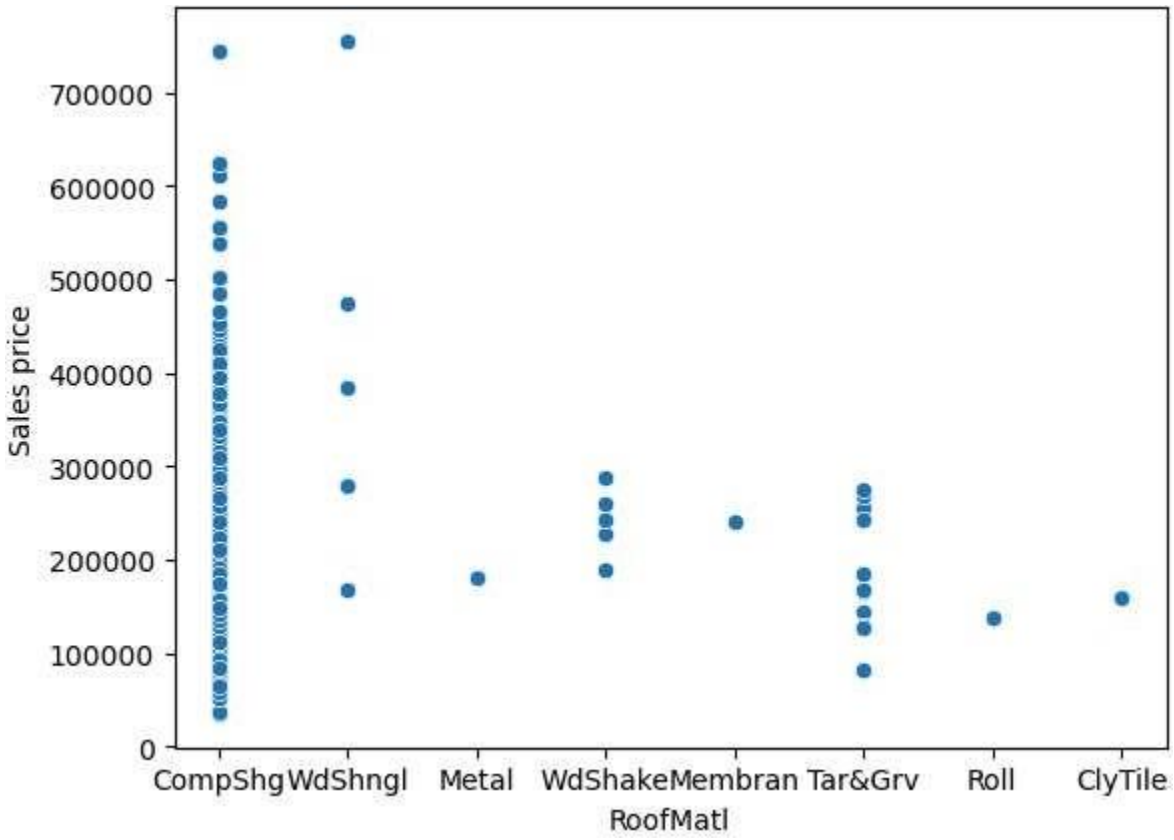

Sales price
700000
600000
500000
400000
300000
200000
100000
0
2
4
6
8
10
OverallQual

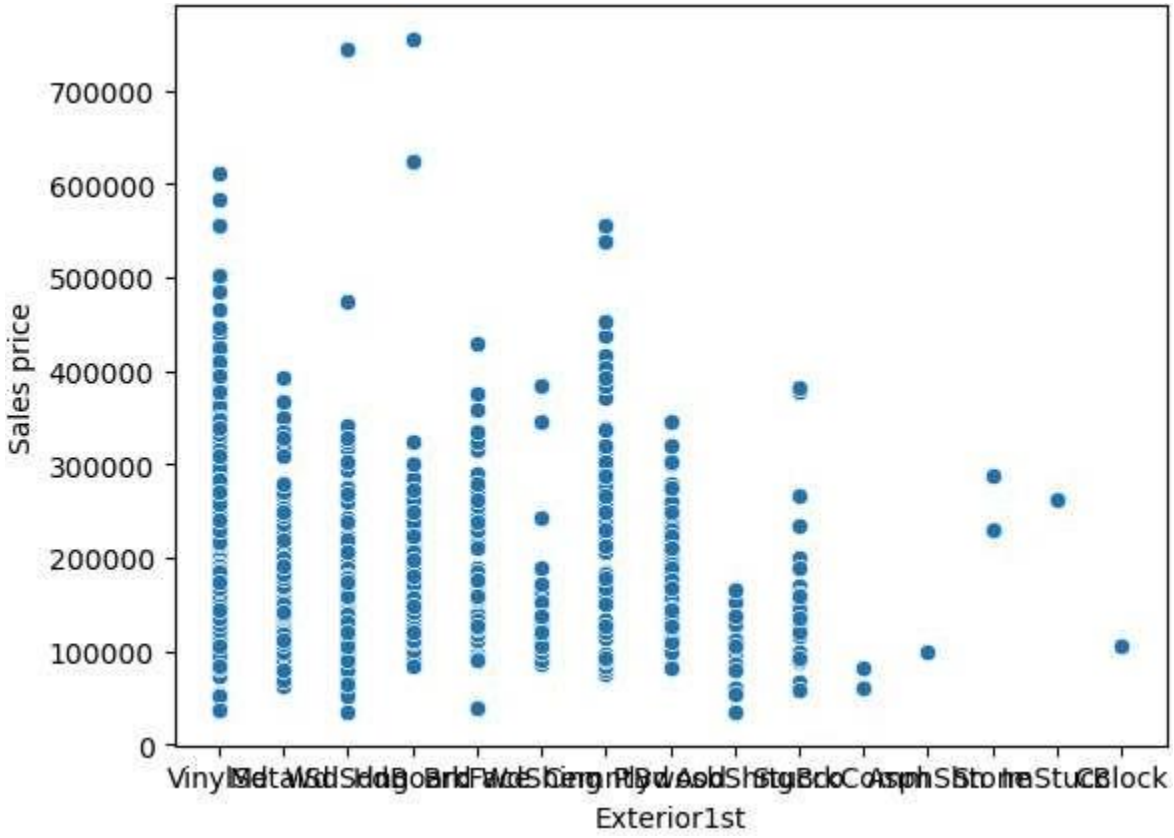

Sales price
0
100000
200000
300000
400000
500000
600000
700000
1
2
3
4
5
6
7
8
9
OverallCond

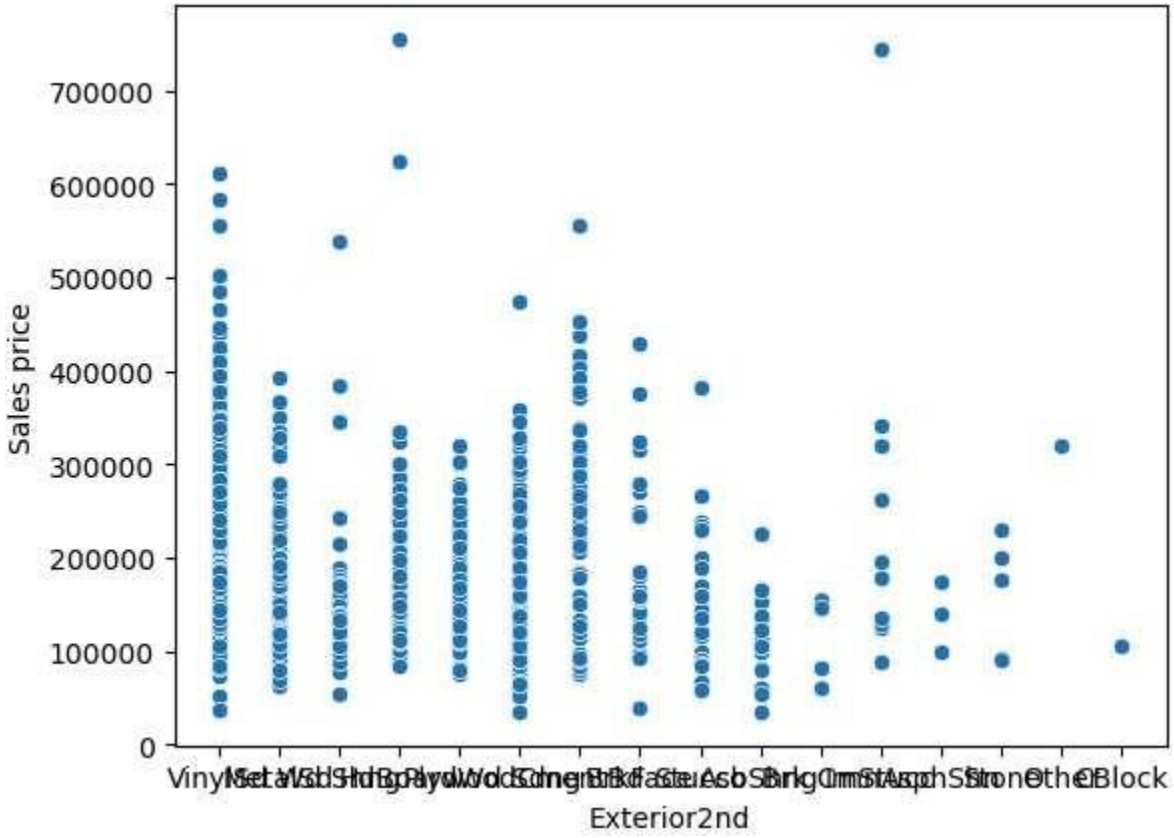

Sales price
700000
600000
500000
400000
300000
200000
100000
0
1880
1900
1920
1940
1960
1980
2000
YearBuilt

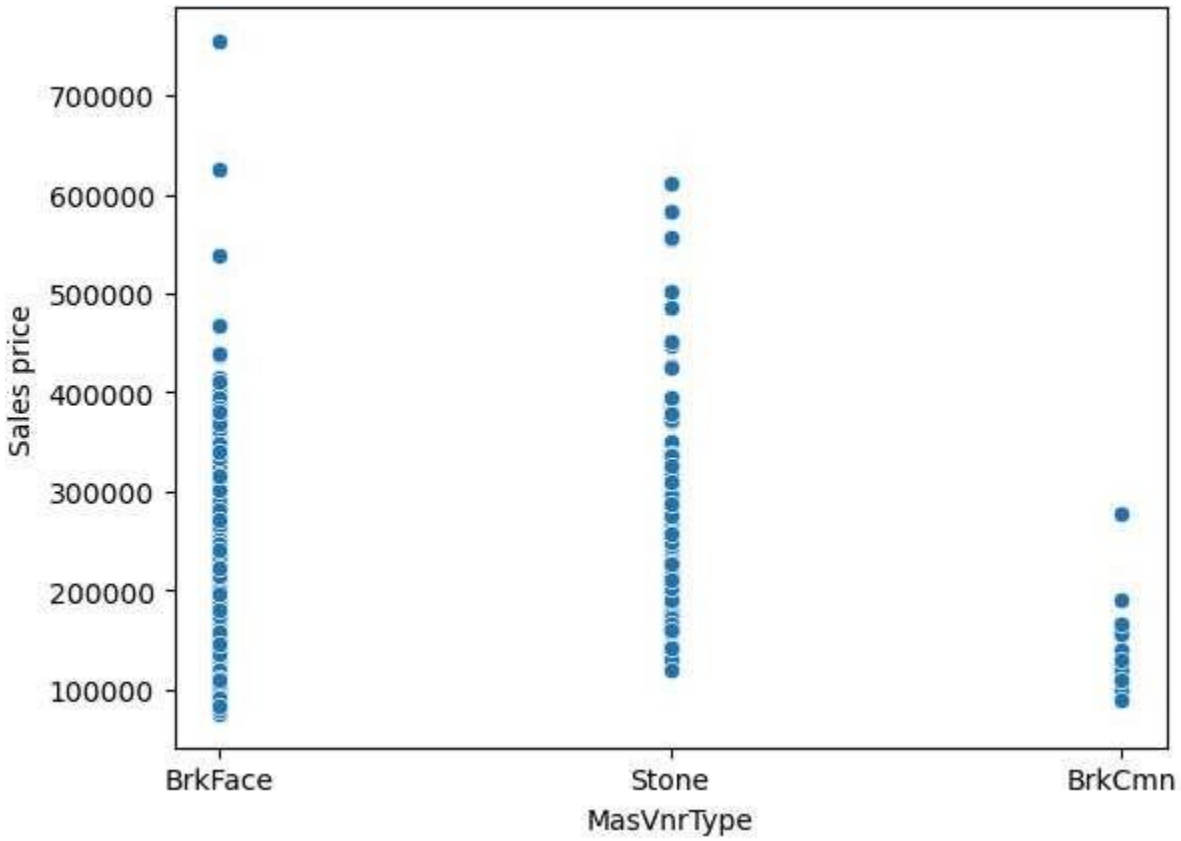

Sales price
700000
600000
500000
400000
300000
200000
100000
0
1950
1960
1970
1980
1990
2000
2010
YearRemodAdd

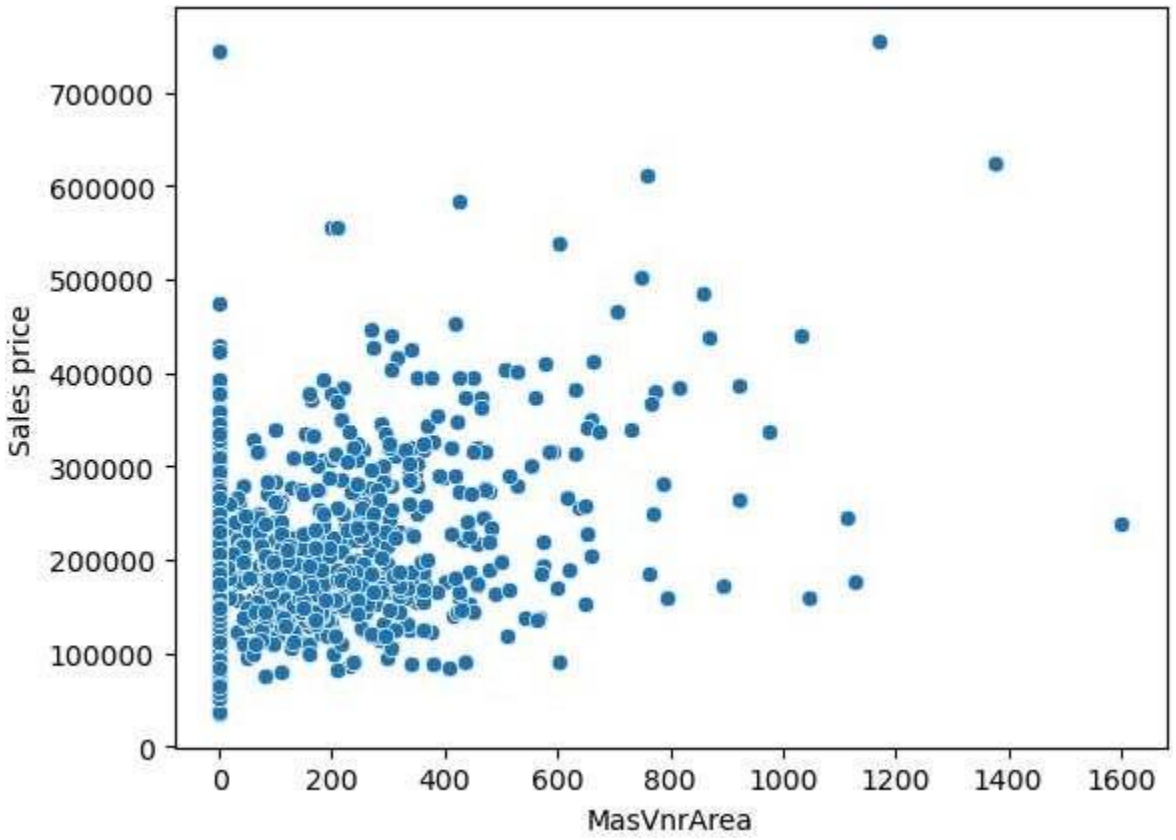

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Gable
Hip
Gambrel
Mansard
Flat
Shed
RoofStyle

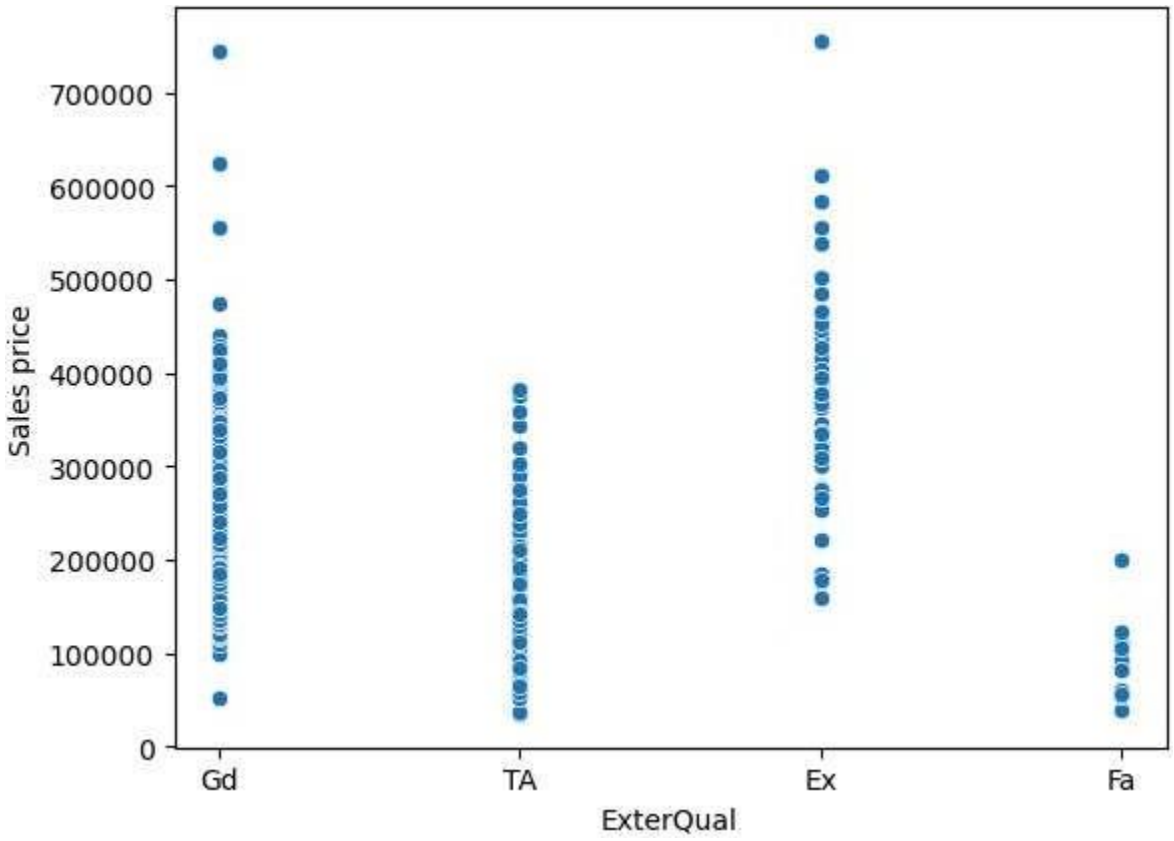

Sales price
700000
600000
500000
400000
300000
200000
100000
0
CompShg
WdShngl
Metal
WdShake
Membran
Tar&Grv
Roll
ClyTile
RoofMatl

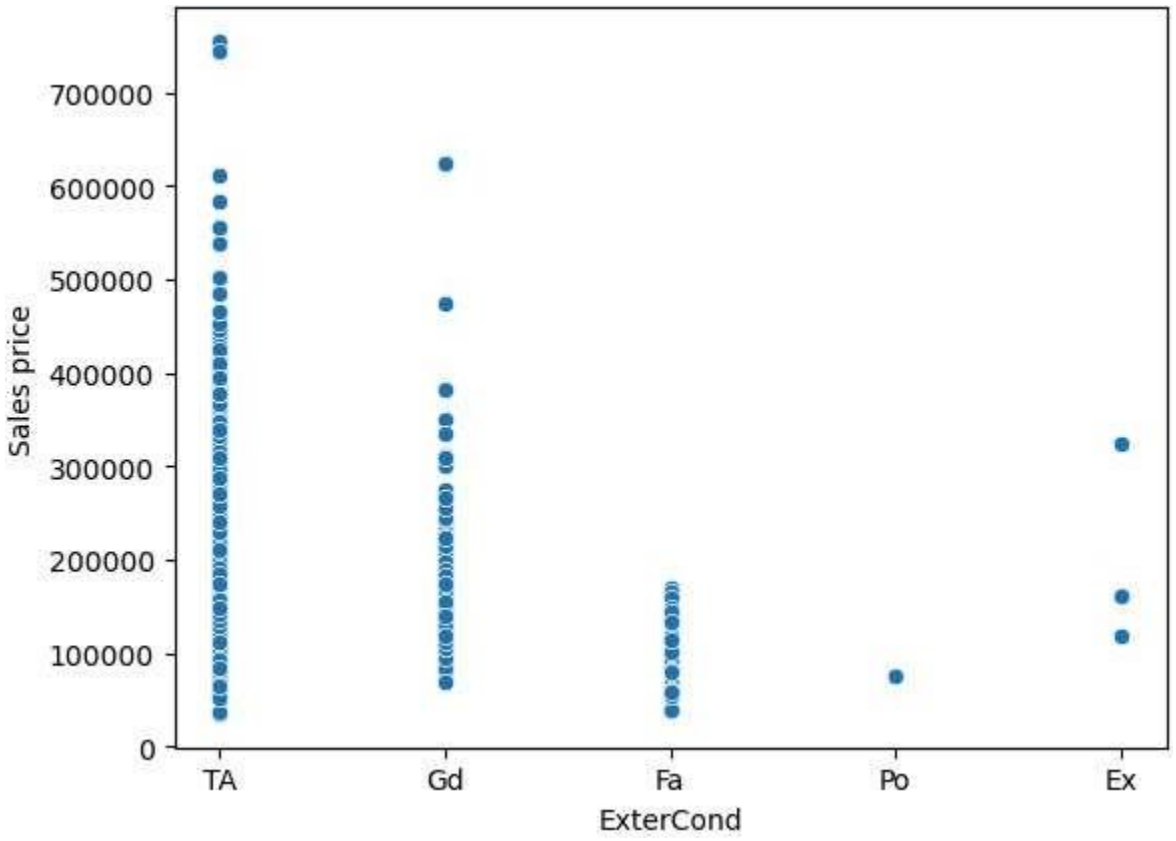

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Exterior1st

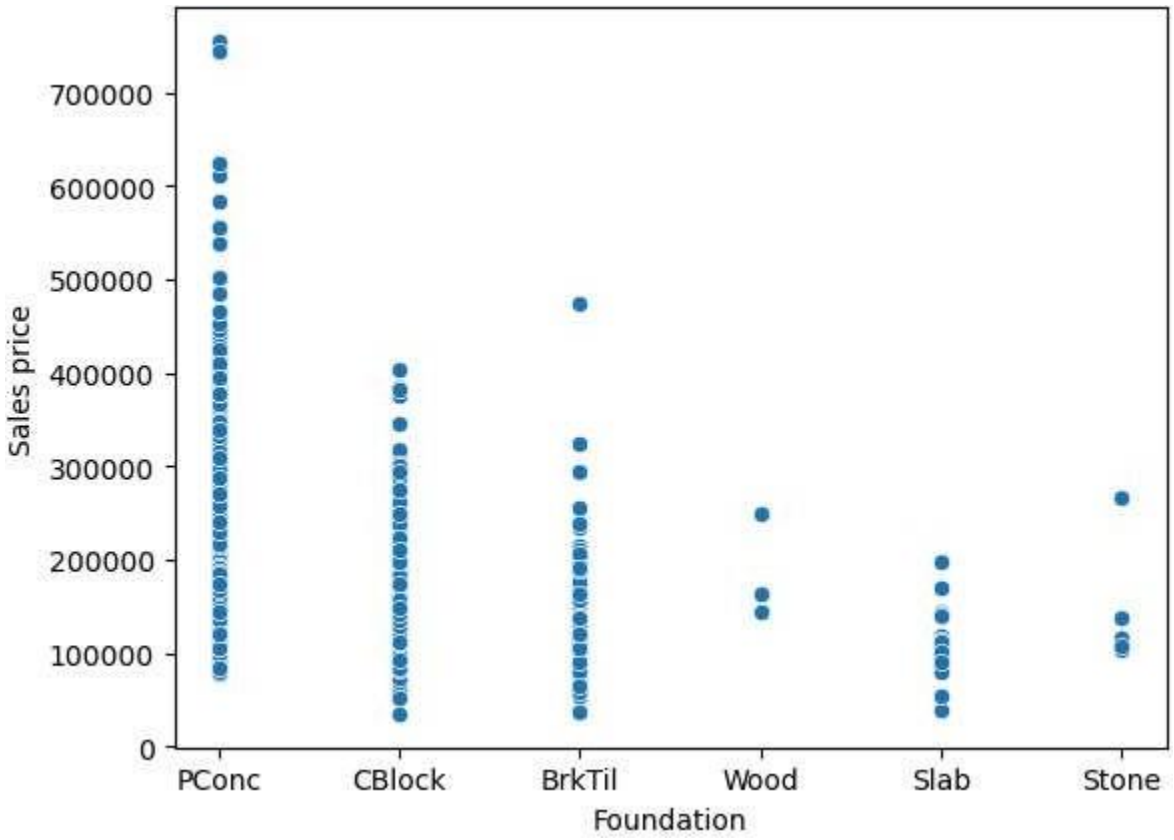

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Exterior2nd

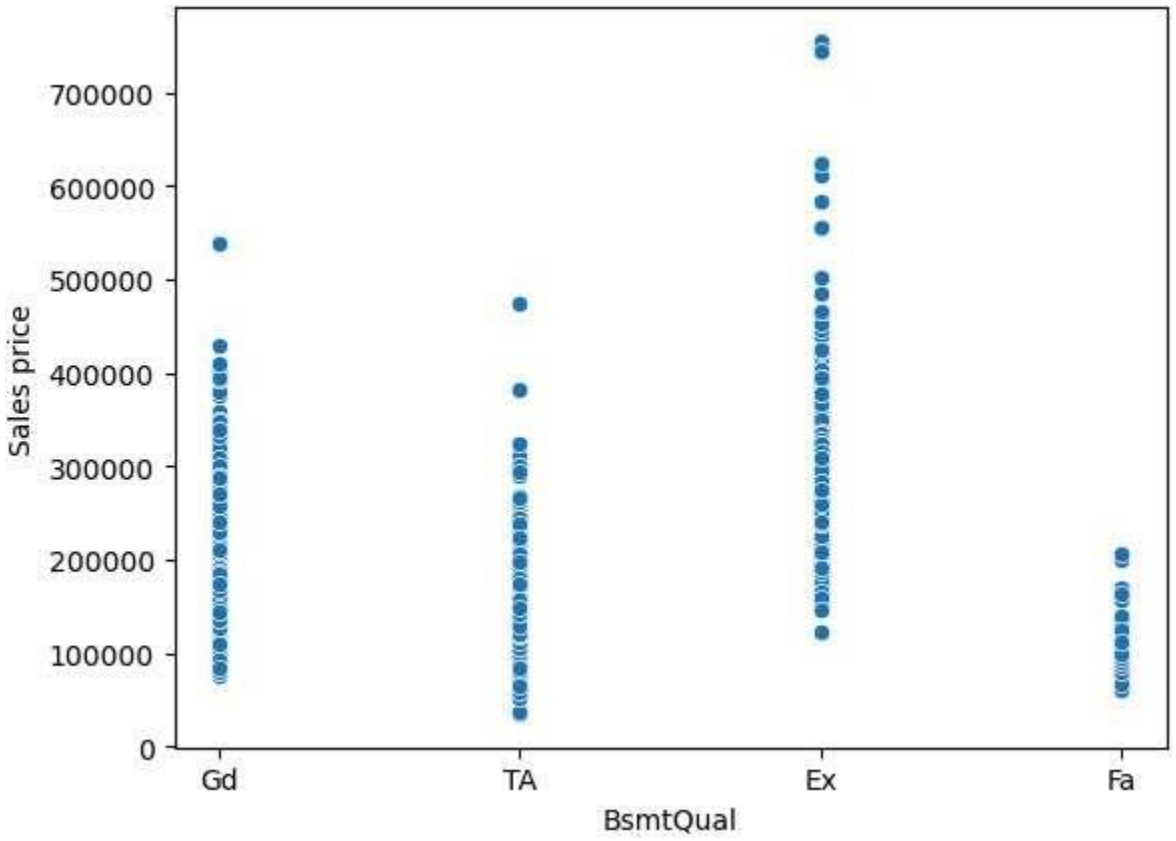

700000
600000
500000
400000
300000
200000
100000
Sales price
BrkFace
Stone
BrkCmn
MasVnrType

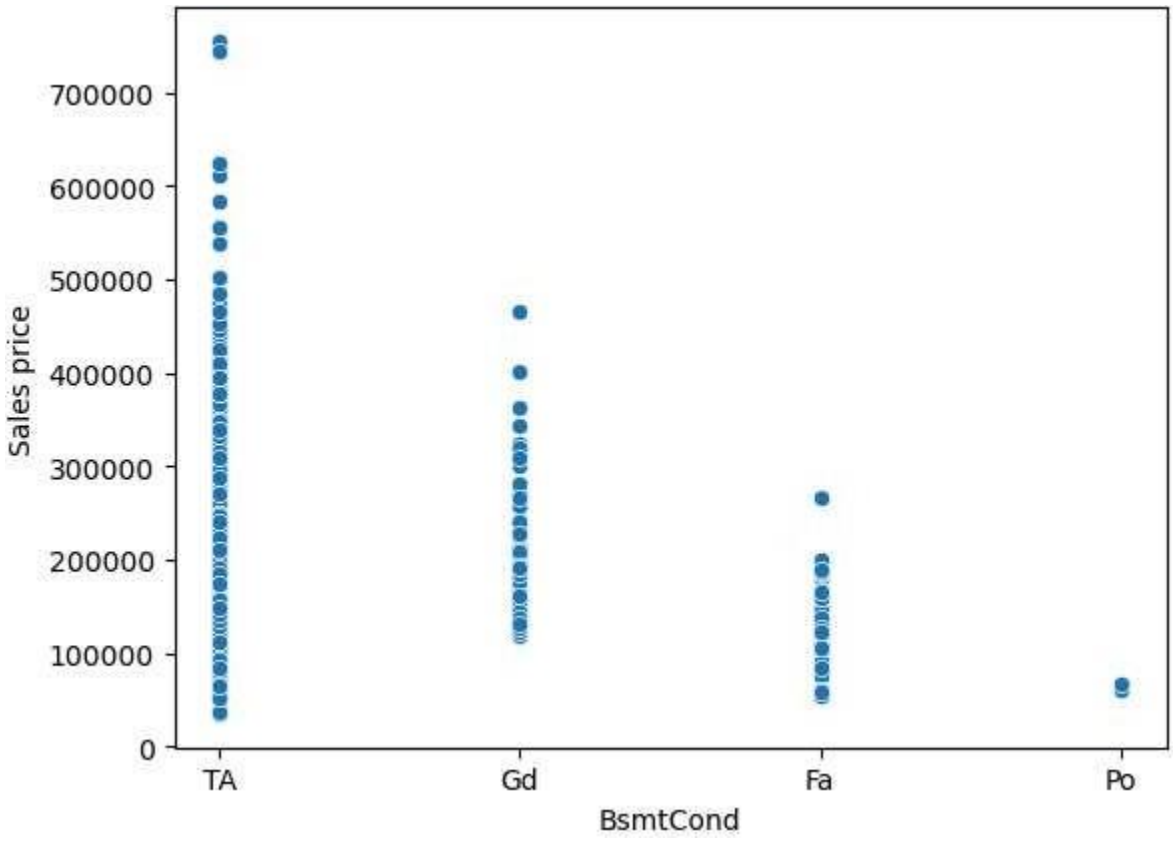

Sales price
MasVnrArea
0
100000
200000
300000
400000
500000
600000
700000
0
200
400
600
800
1000
1200
1400
1600

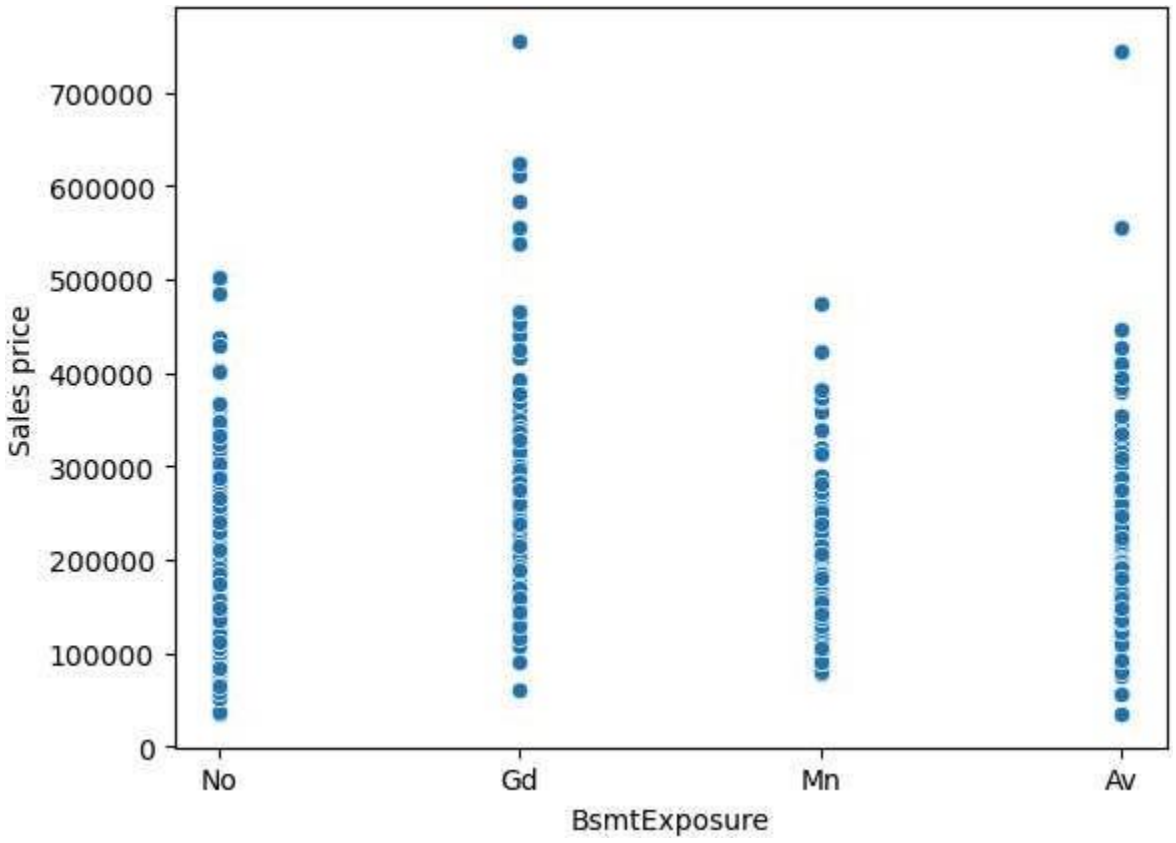

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Gd
TA
Ex
Fa
ExterQual

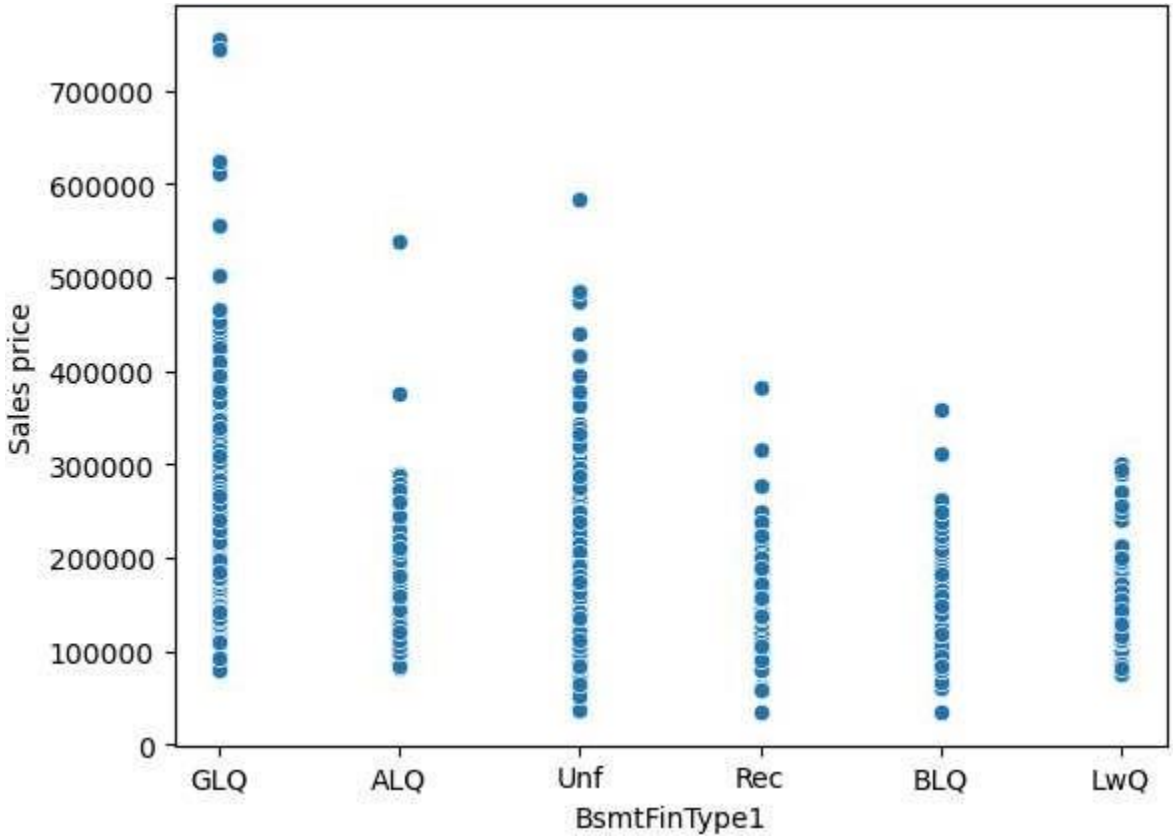

Sales price
700000
600000
500000
400000
300000
200000
100000
0
TA
Gd
Fa
Po
Ex
ExterCond

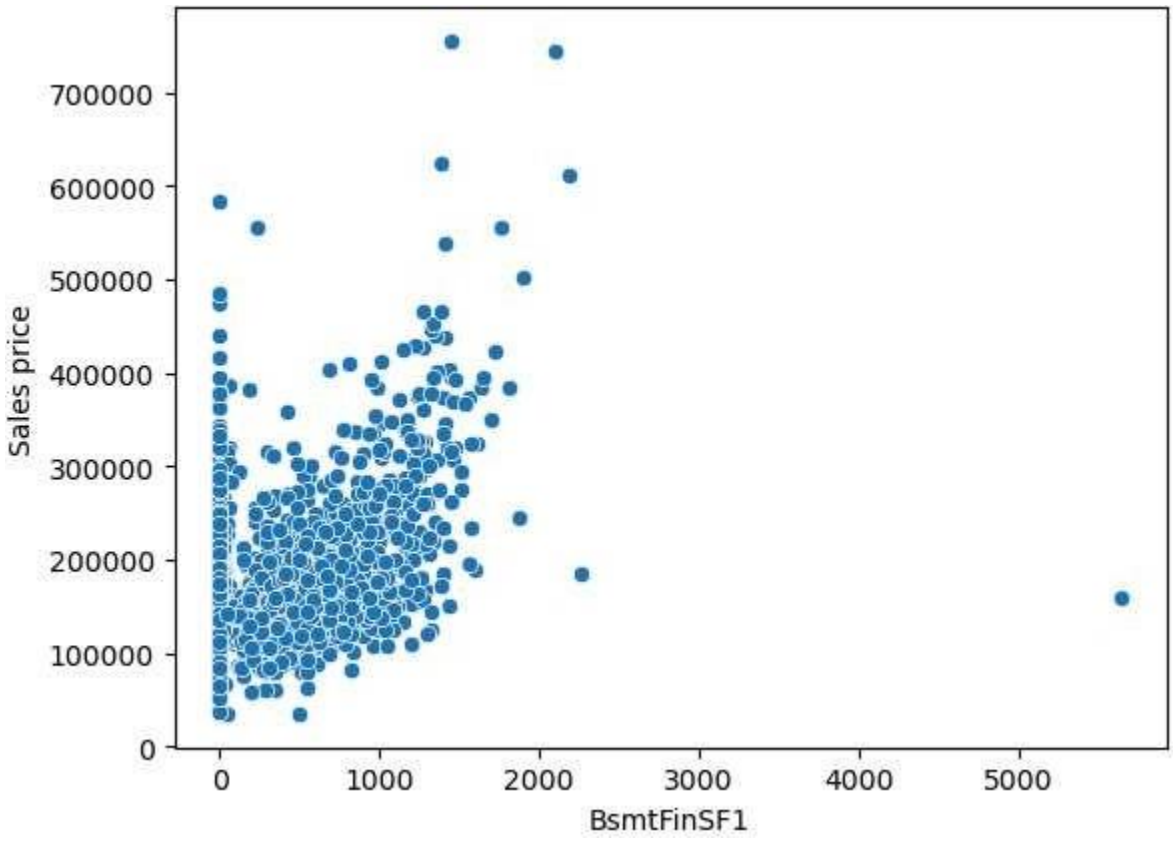

Sales price
700000
600000
500000
400000
300000
200000
100000
0
PConc
CBlock
BrkTil
Wood
Slab
Stone
Foundation

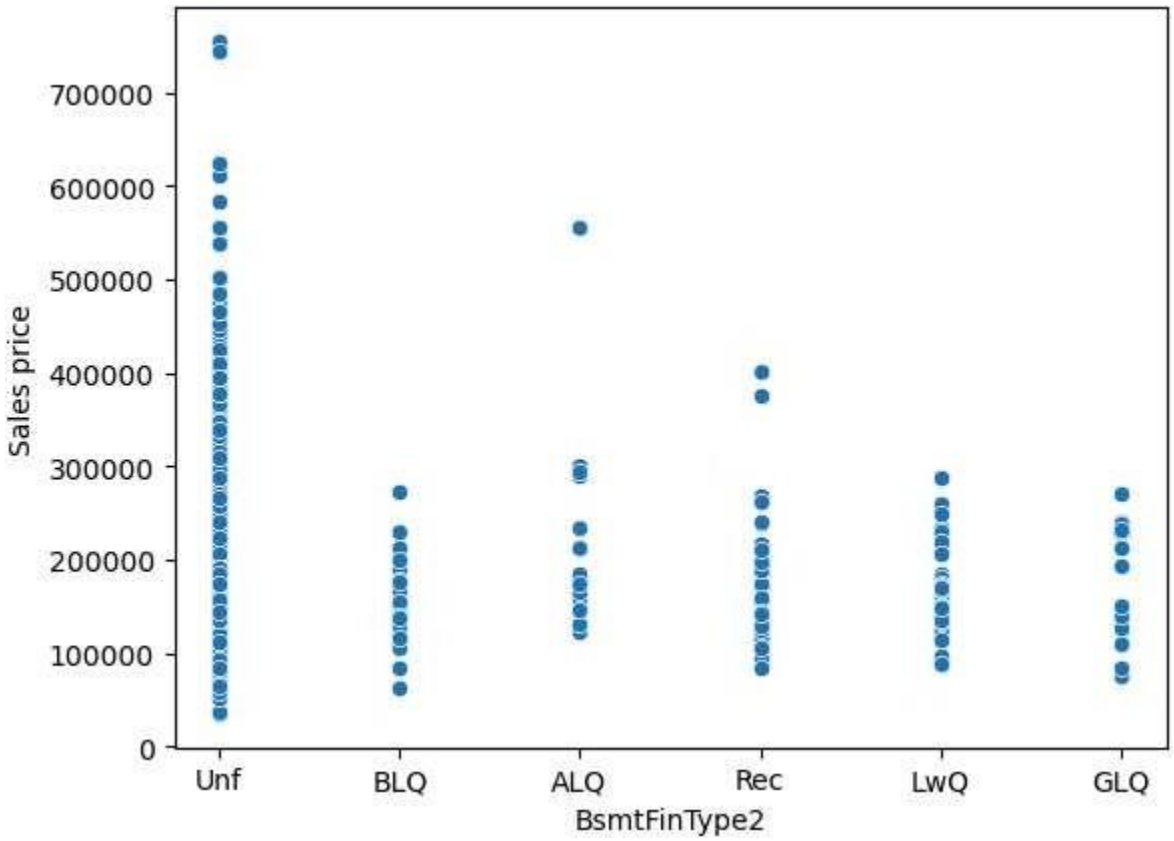

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Gd
TA
Ex
Fa
BsmtQual

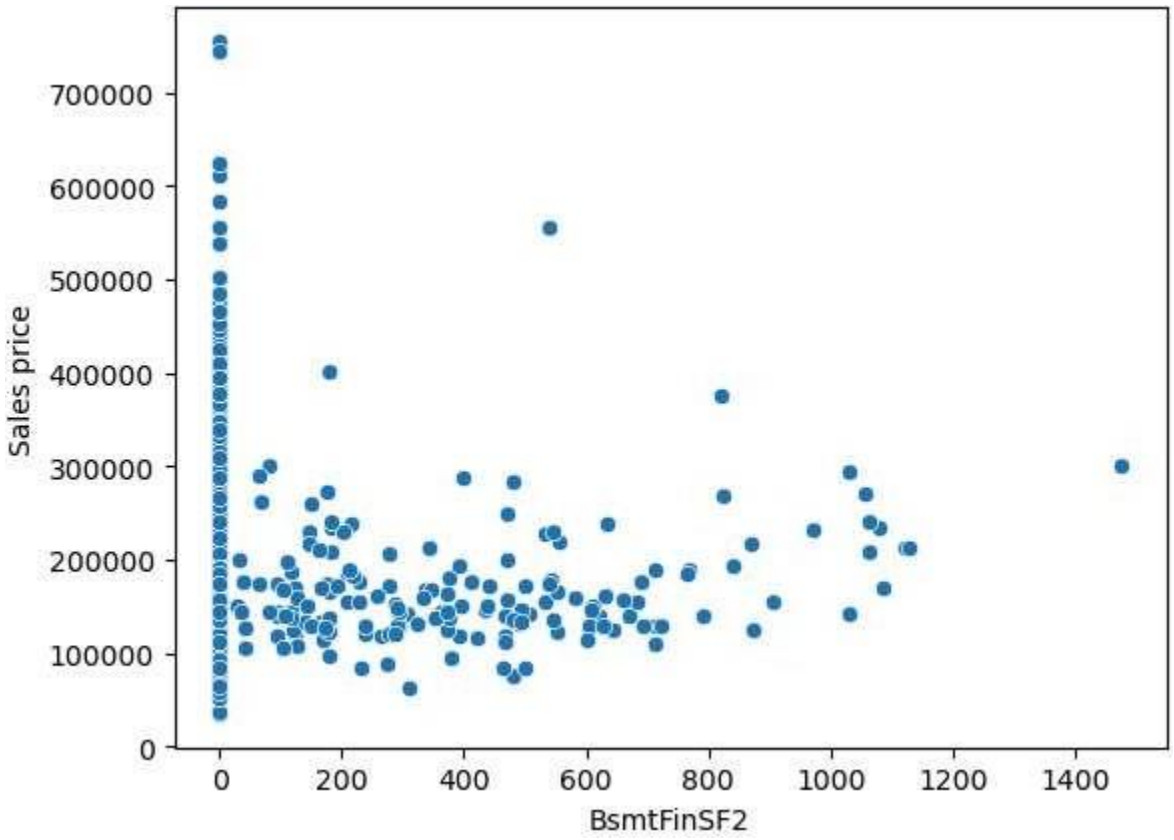

Sales price
700000
600000
500000
400000
300000
200000
100000
0
TA
Gd
Fa
Po
BsmtCond

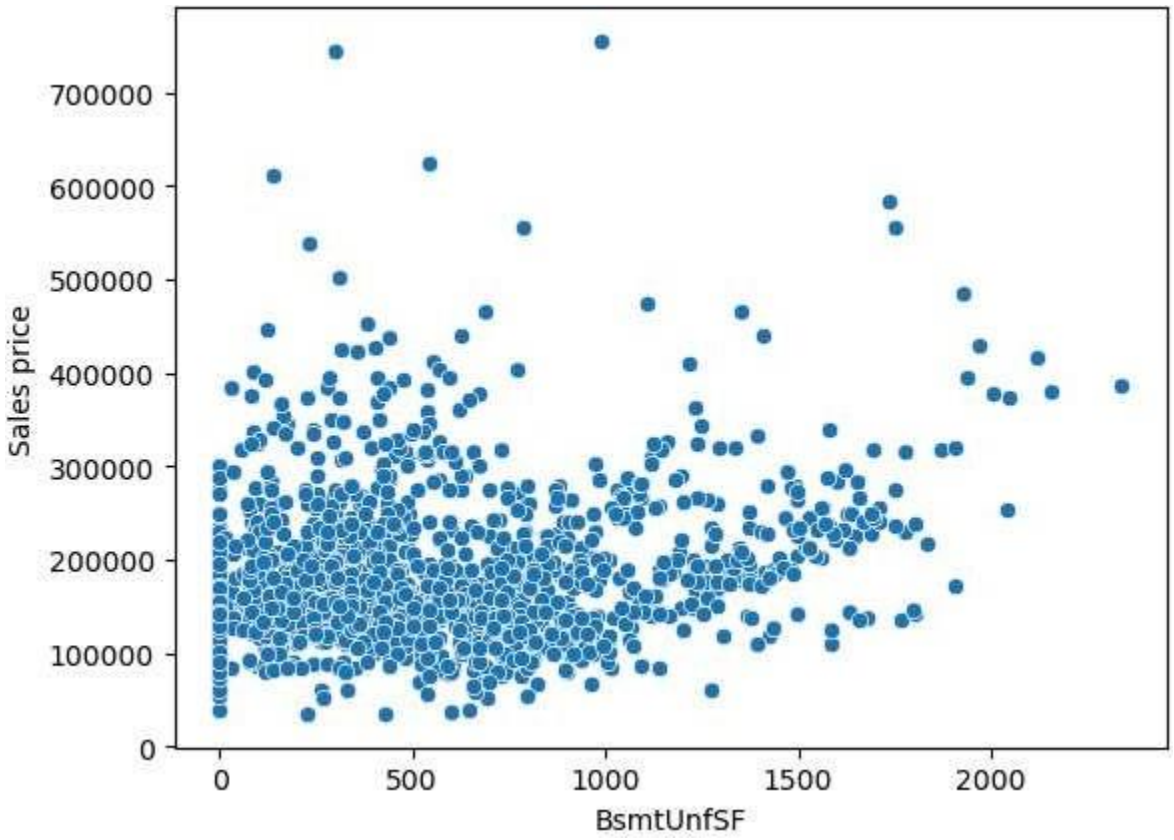

700000
600000
500000
400000
300000
200000
100000
0
Sales price
No
Gd
Mn
Av
BsmtExposure

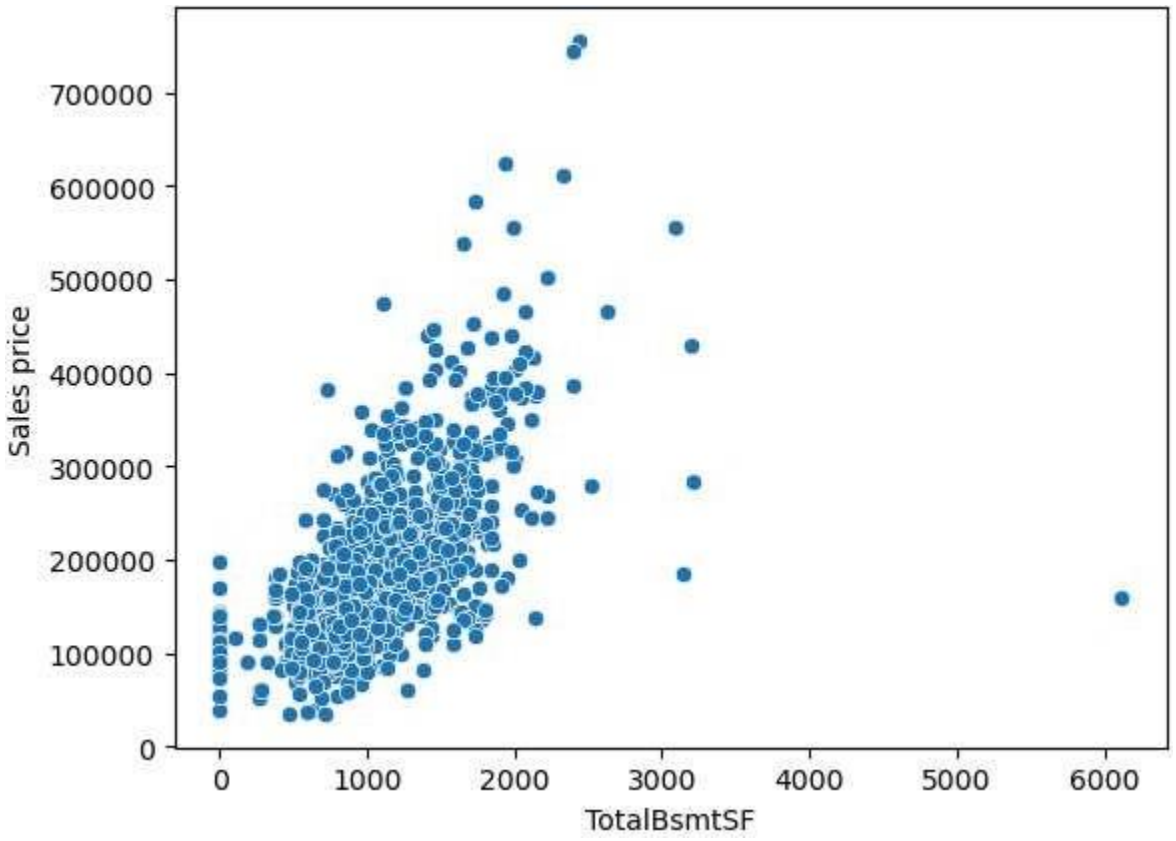

700000
600000
500000
400000
300000
200000
100000
0
Sales price
GLQ
ALQ
Unf
Rec
BLQ
LwQ
BsmtFinType1

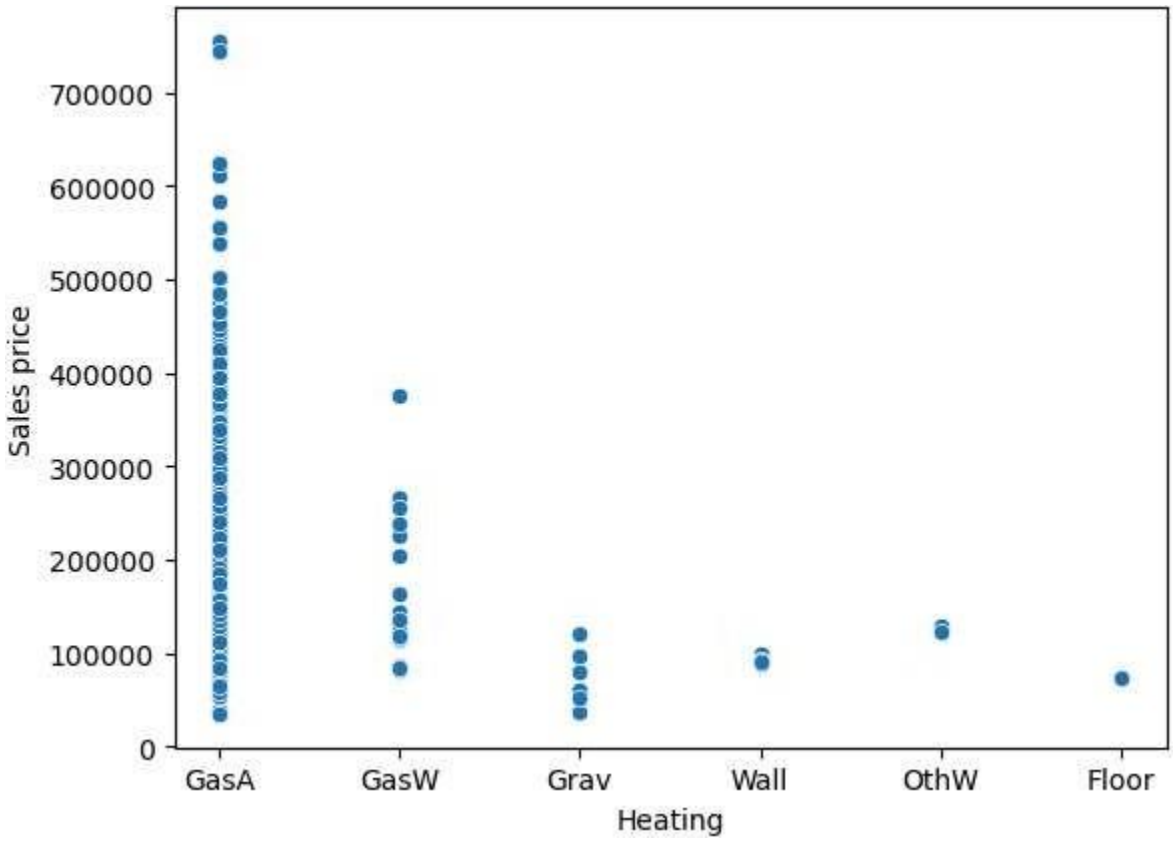

Sales price
BsmtFinSF1
0
100000
200000
300000
400000
500000
600000
700000
0
1000
2000
3000
4000
5000

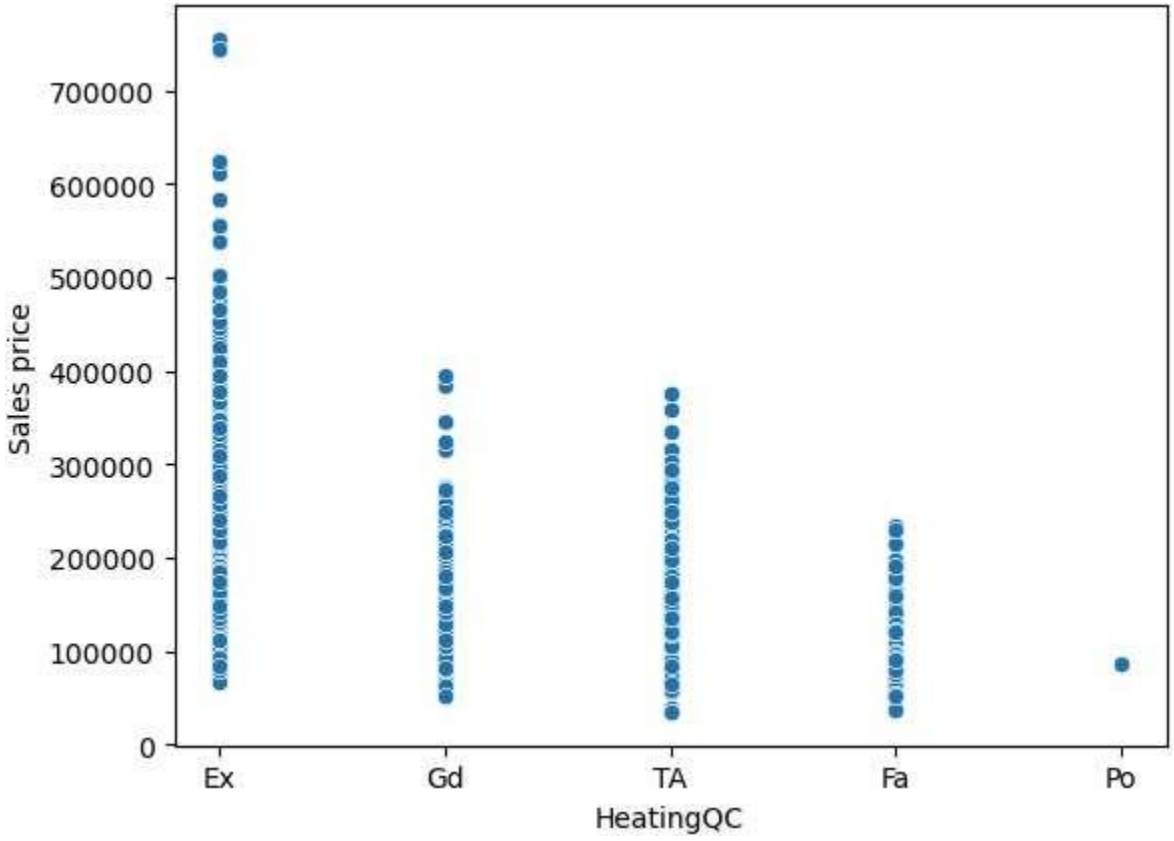

Sales price
700000
600000
500000
400000
300000
200000
100000
0
Unf
BLQ
ALQ
Rec
LwQ
GLQ
BsmtFinType2

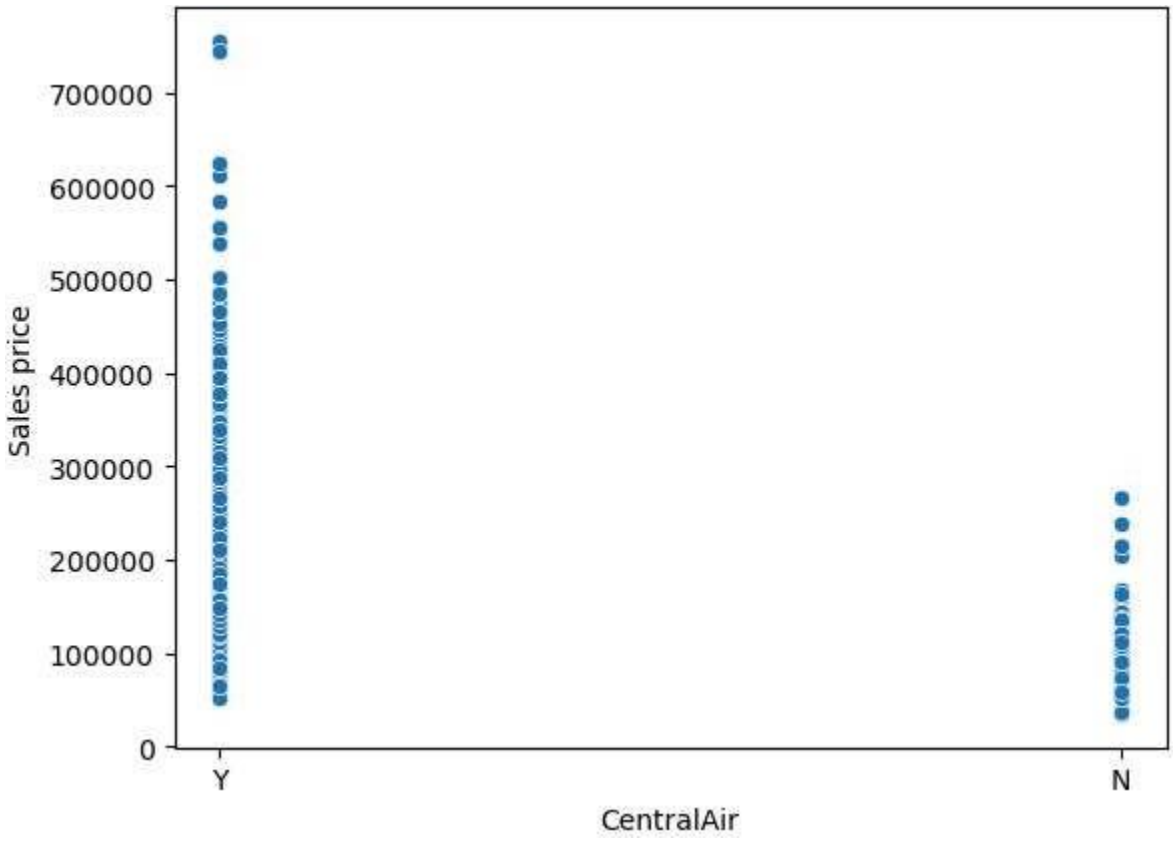

Sales price
700000
600000
500000
400000
300000
200000
100000
0
0
200
400
600
800
1000
1200
1400
BsmtFinSF2

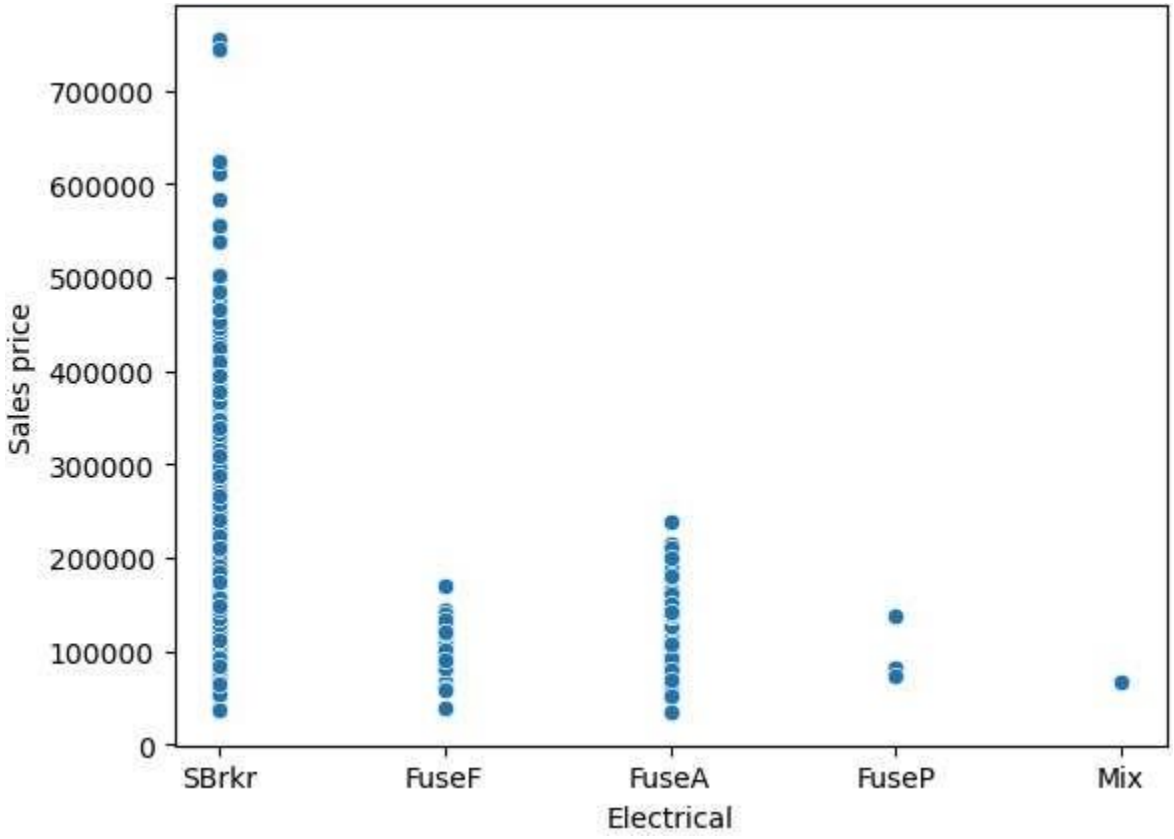

Sales price
700000
600000
500000
400000
300000
200000
100000
0
0
500
1000
1500
2000
BsmtUnfSF

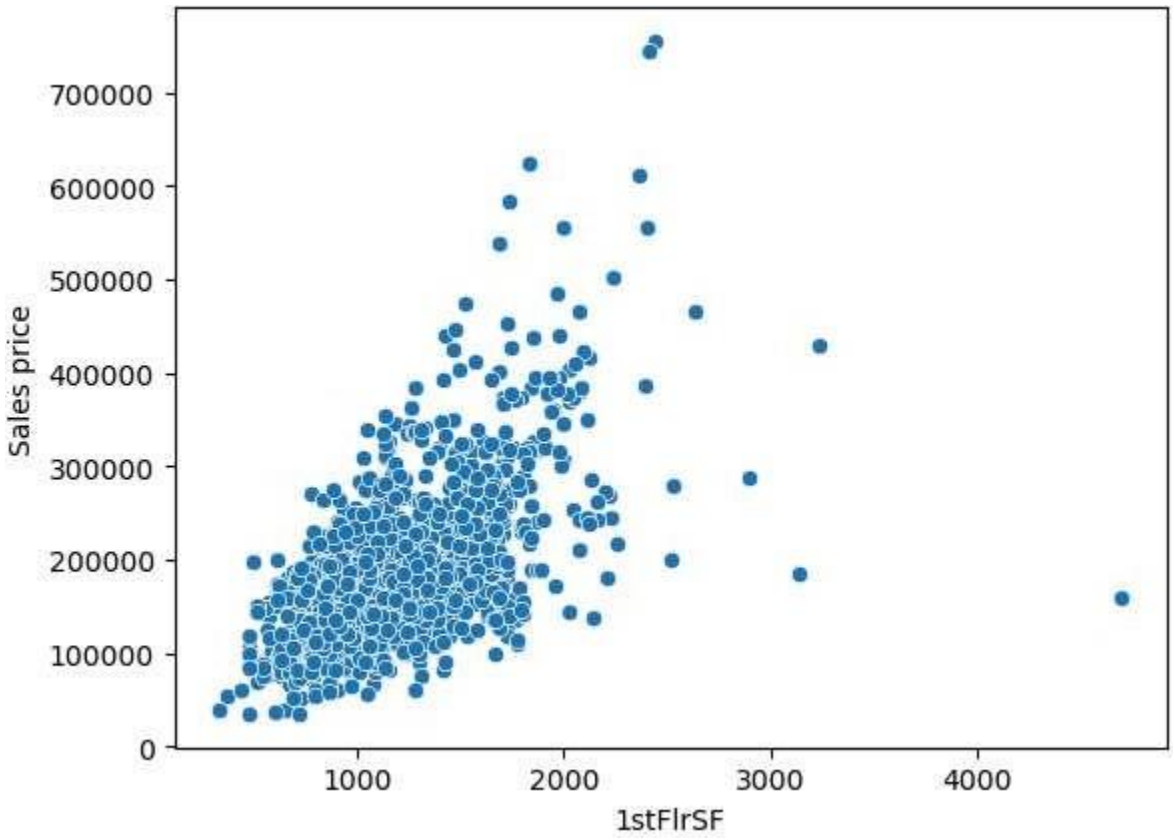

Sales price
700000
600000
500000
400000
300000
200000
100000
0
0
1000
2000
3000
4000
5000
6000
TotalBsmtSF

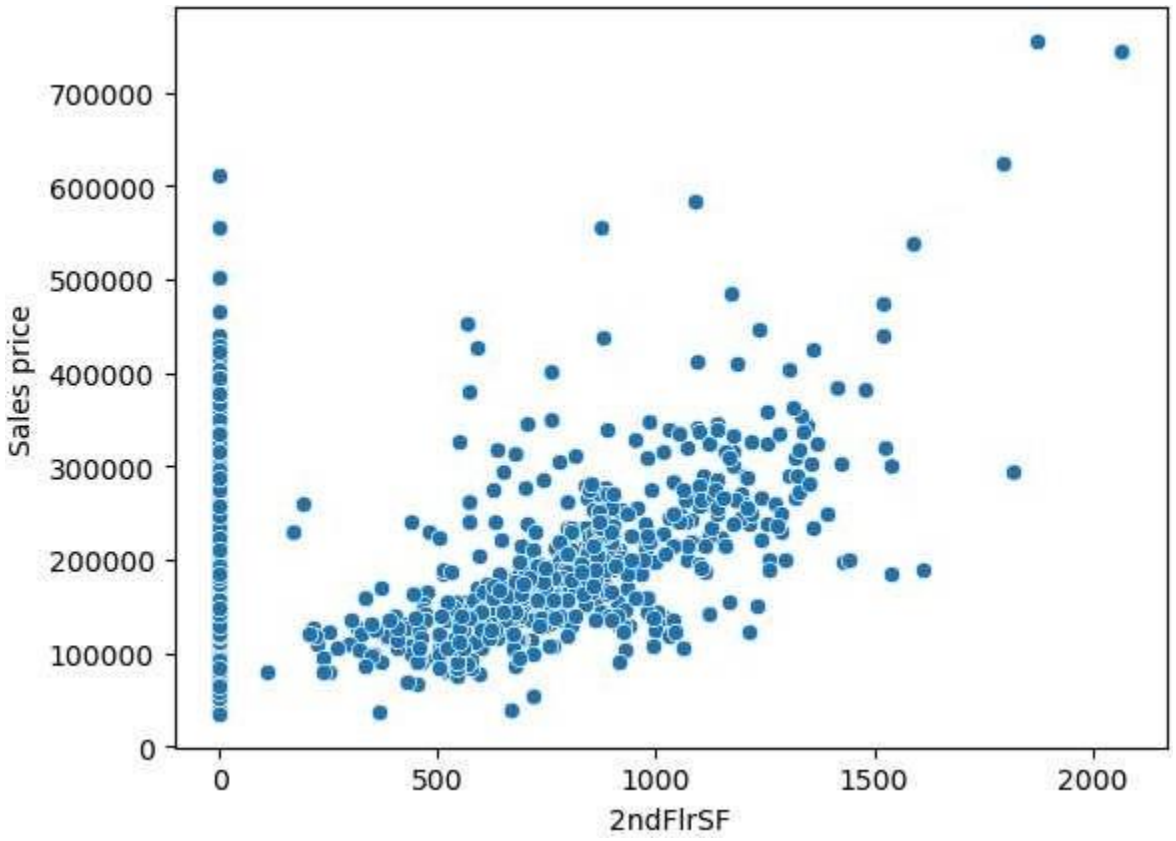

Sales price
700000
600000
500000
400000
300000
200000
100000
0
GasA
GasW
Grav
Wall
OthW
Floor
Heating

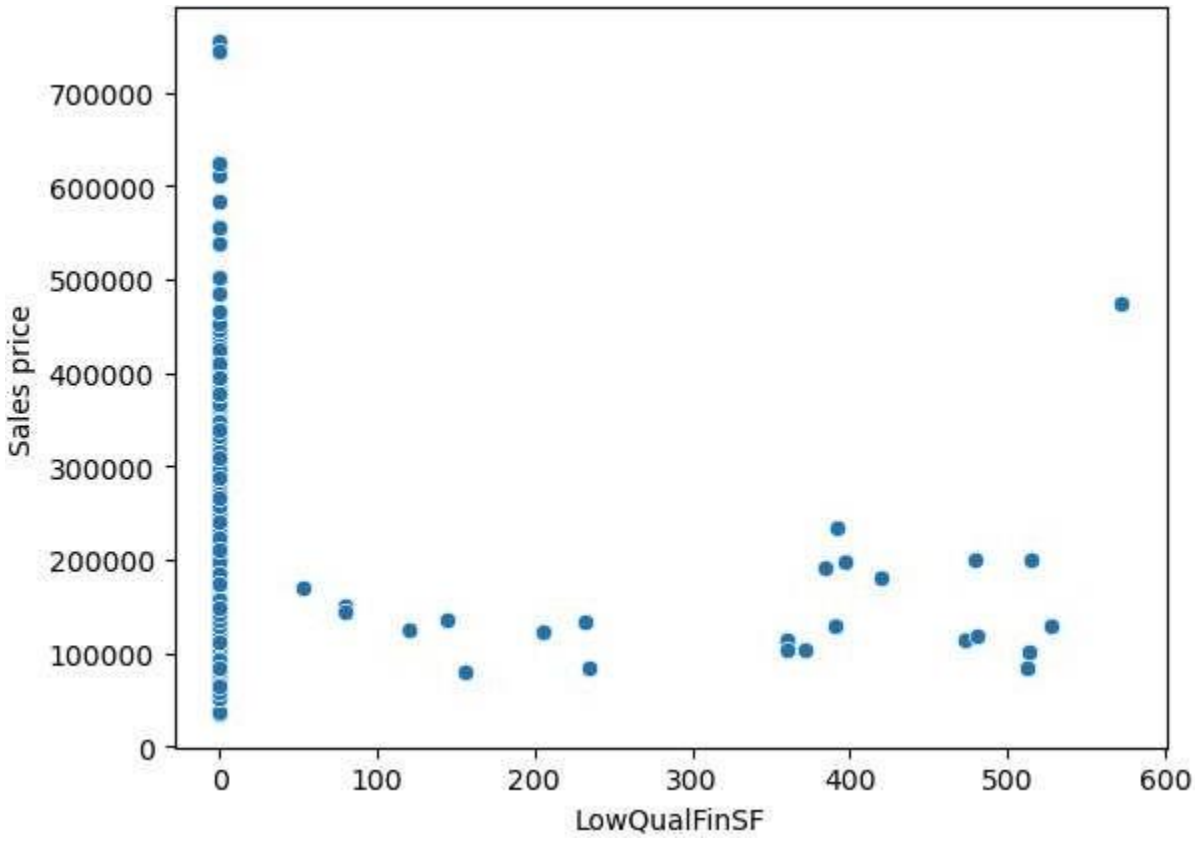

Sales price
700000
600000
500000
400000
300000
200000
100000
0
Ex
Gd
TA
Fa
Po
HeatingQC

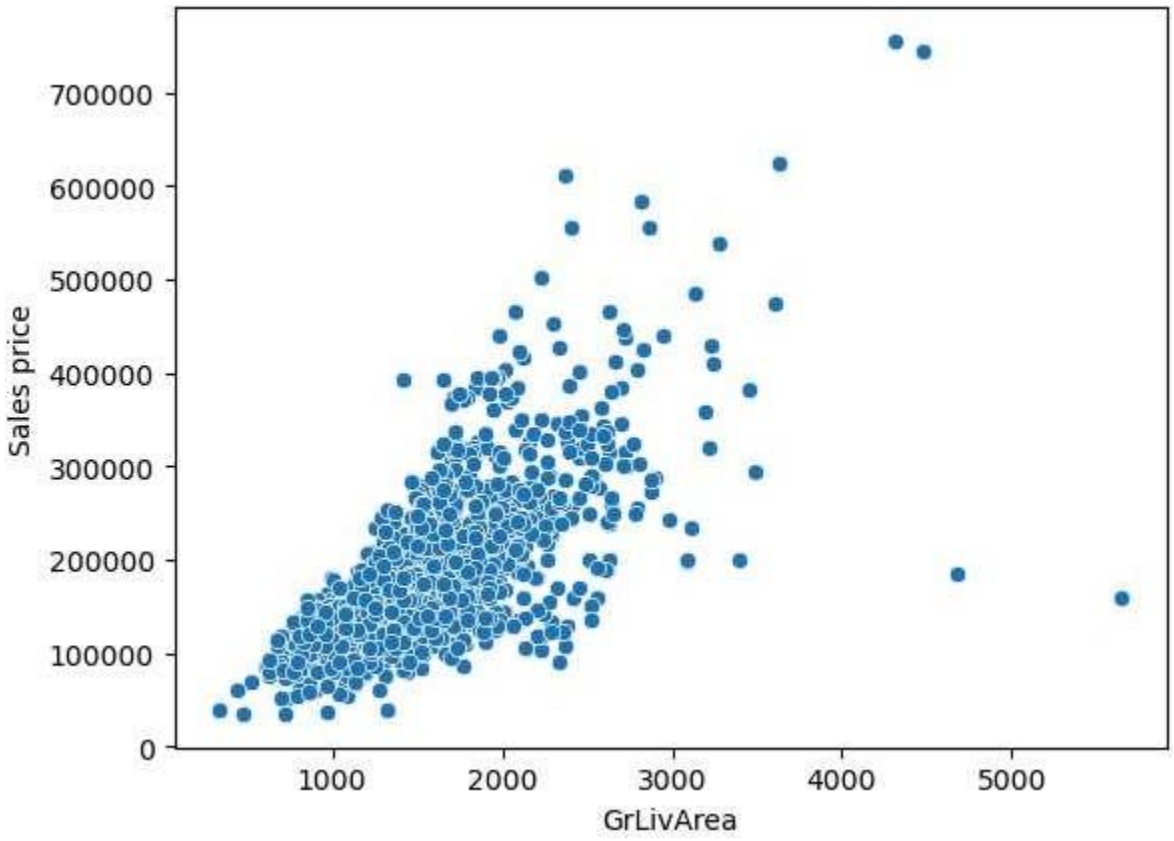

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Y
N
CentralAir

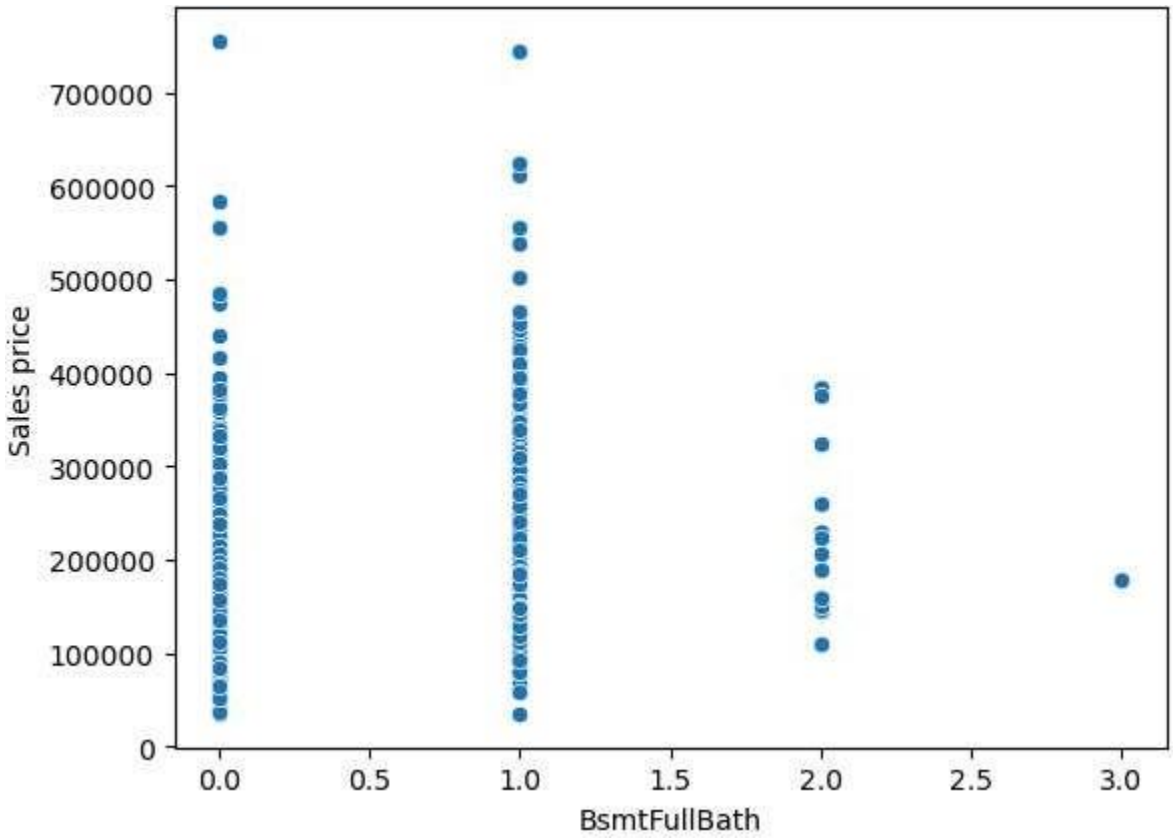

Sales price
700000
600000
500000
400000
300000
200000
100000
0
SBrkr
FuseF
FuseA
FuseP
Mix
Electrical

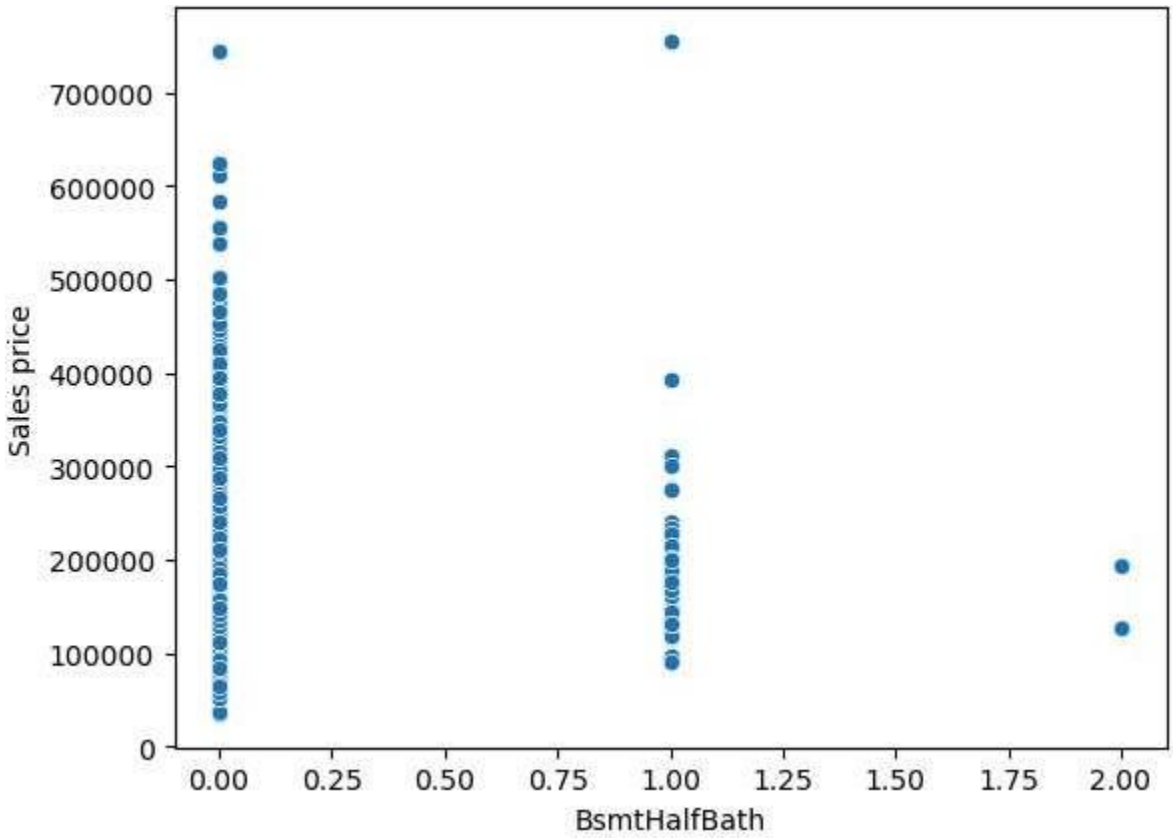

Sales price
700000
600000
500000
400000
300000
200000
100000
0
1000
2000
3000
4000
1stFlrSF

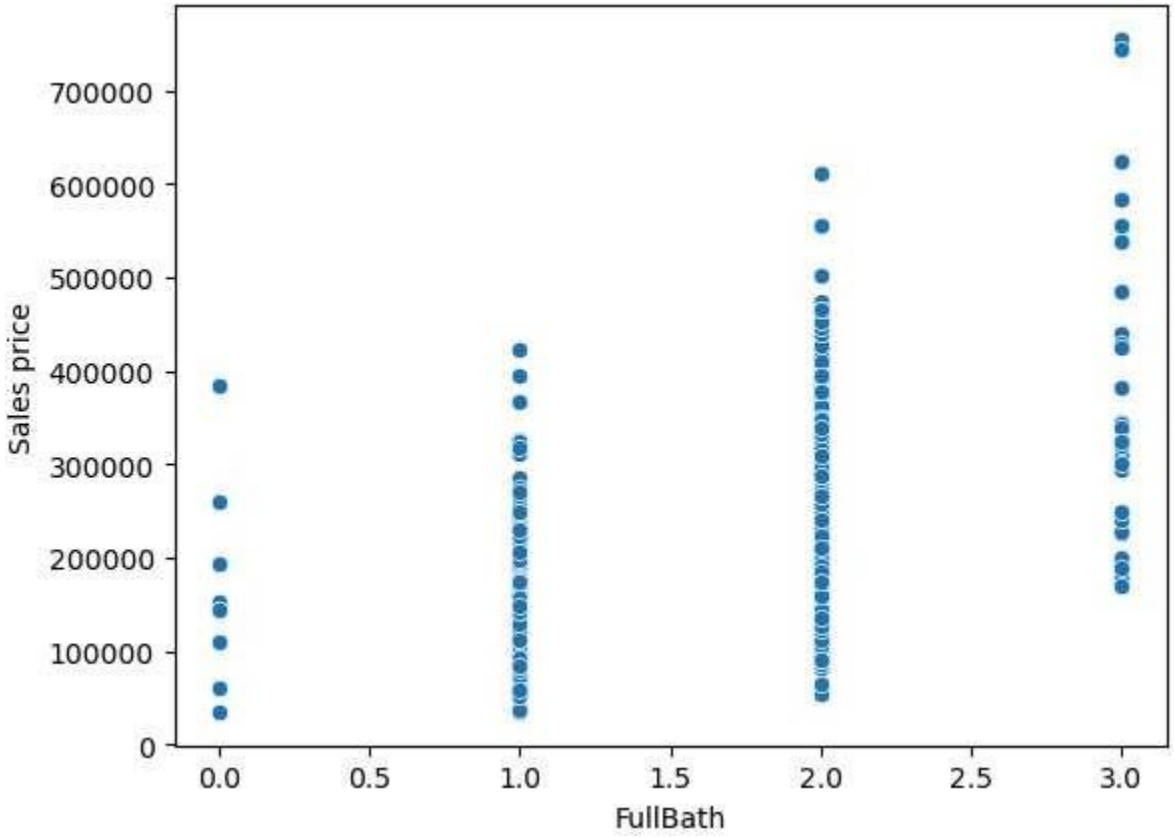

Sales price
700000
600000
500000
400000
300000
200000
100000
0
0
500
1000
1500
2000
2ndFlrSF

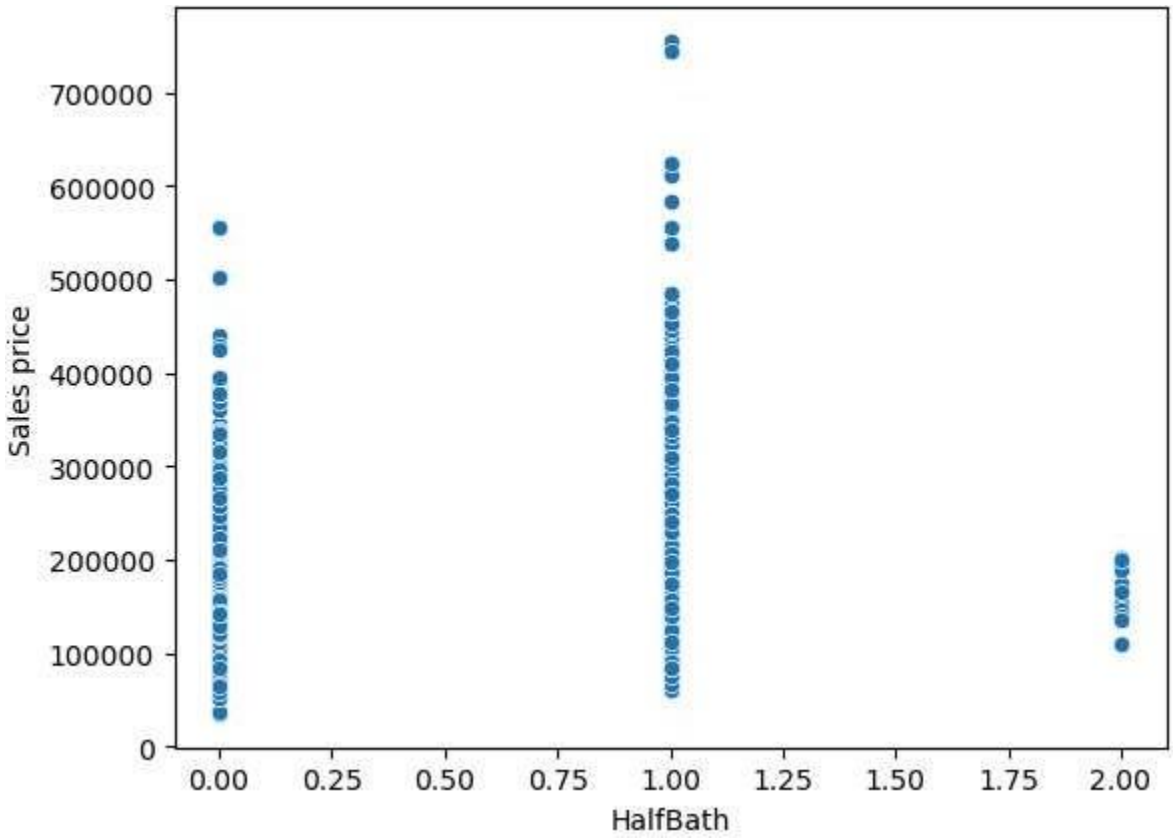

Sales price
LowQualFinSF
0
100000
200000
300000
400000
500000
600000
700000
0
100
200
300
400
500
600

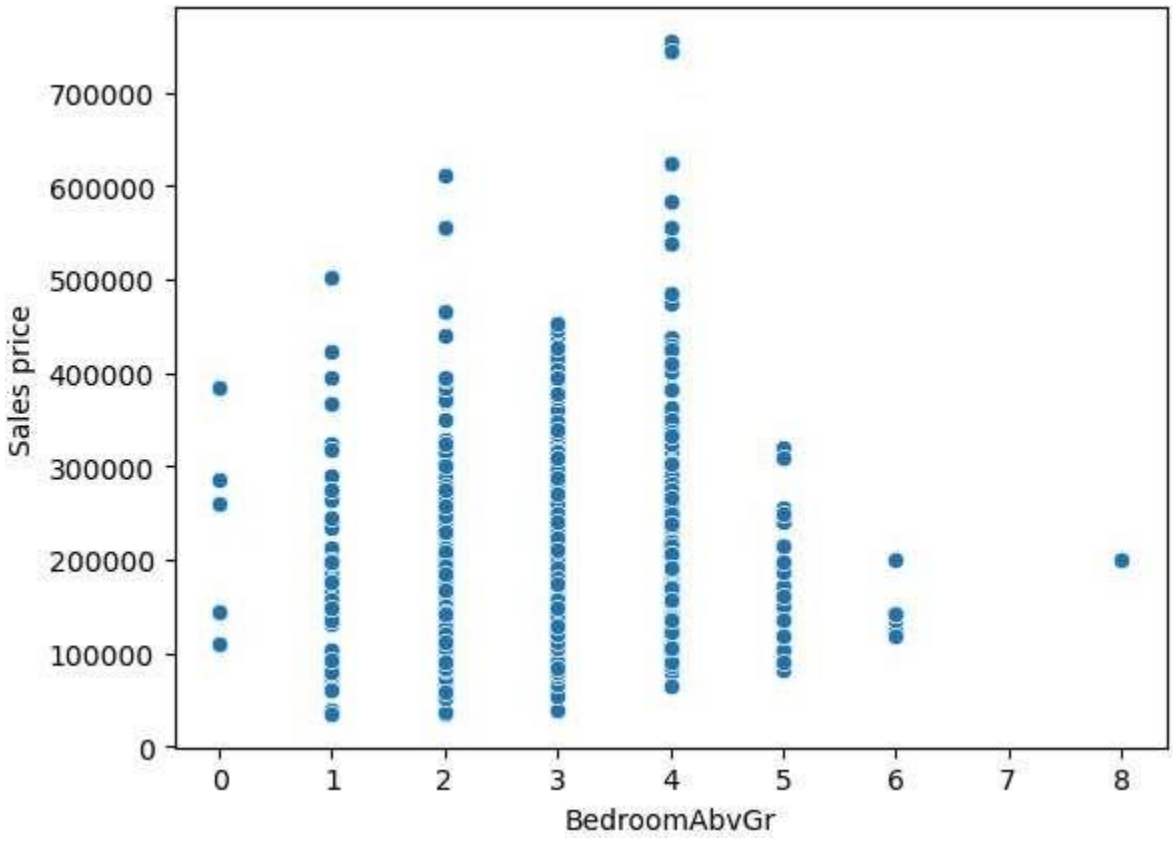

Sales price
700000
600000
500000
400000
300000
200000
100000
0
1000
2000
3000
4000
5000
GrLivArea

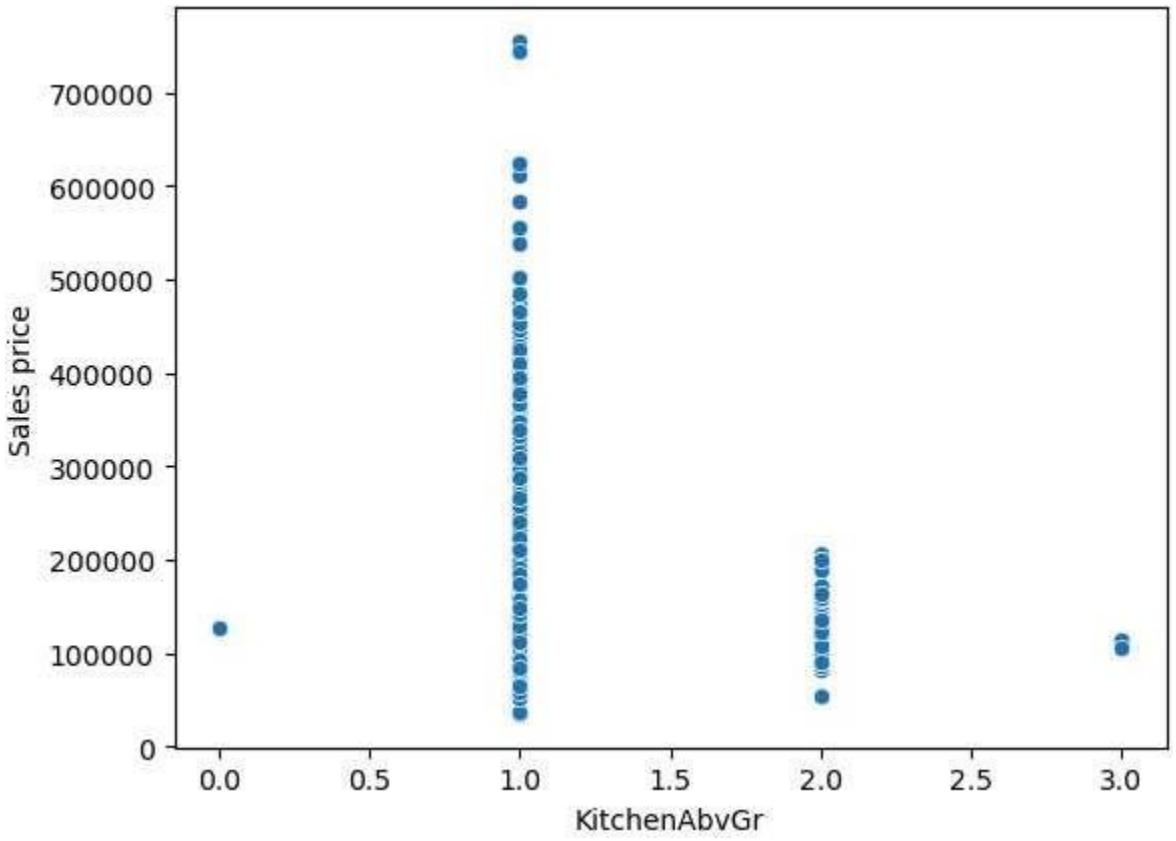

Sales price
700000
600000
500000
400000
300000
200000
100000
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
BsmtFullBath

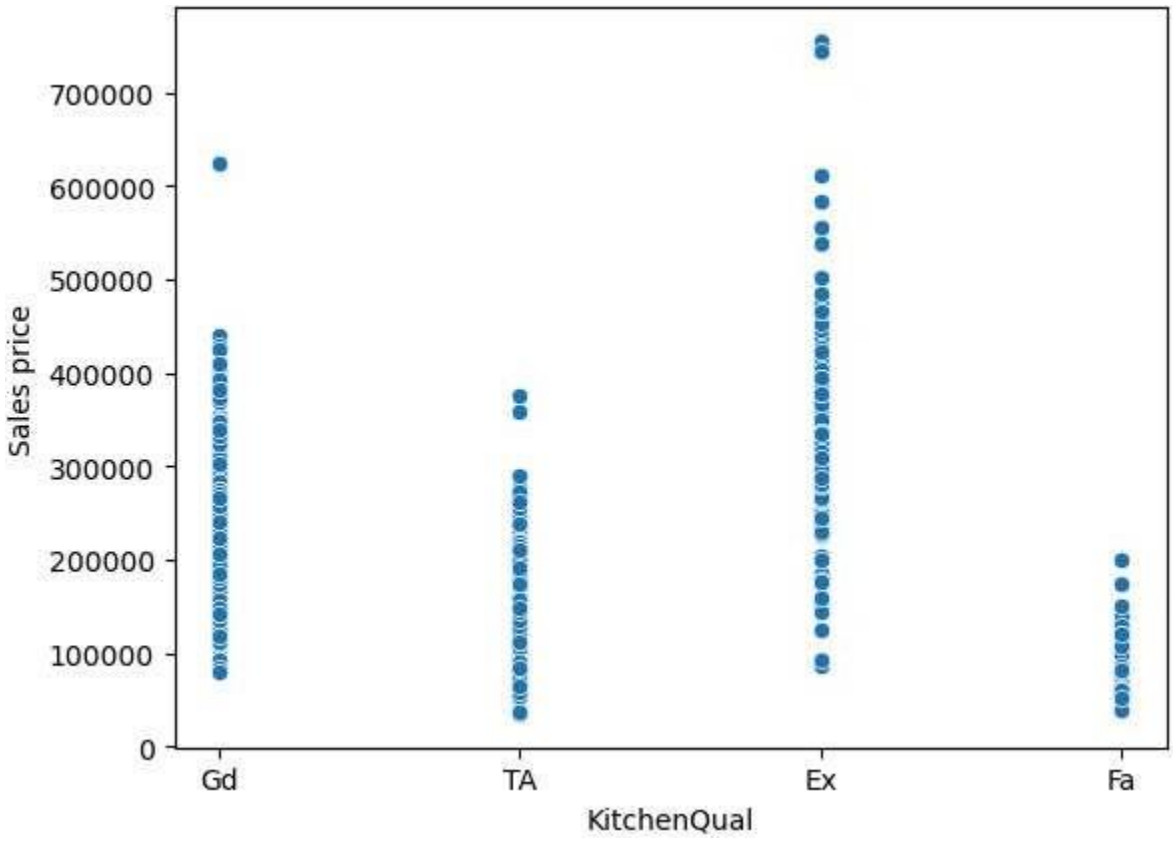

Sales price
700000
600000
500000
400000
300000
200000
100000
0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
BsmtHalfBath

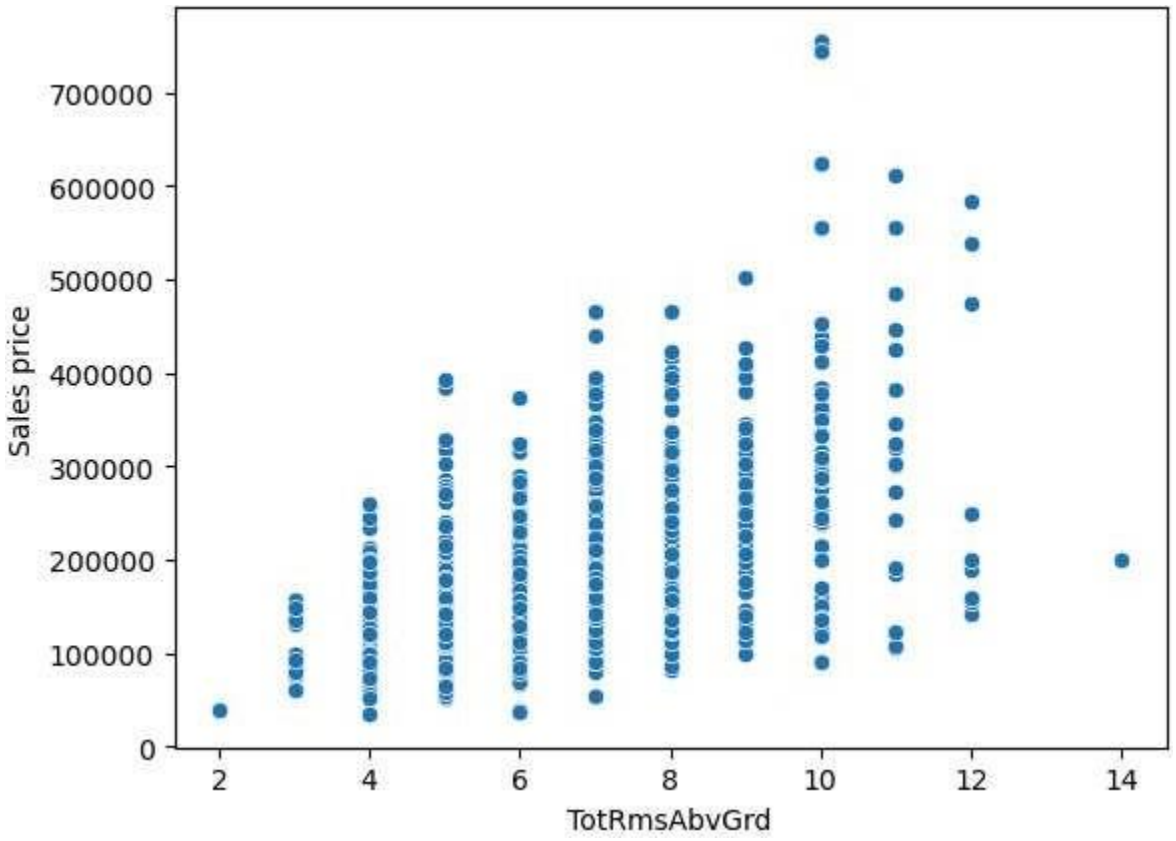

Sales price
700000
600000
500000
400000
300000
200000
100000
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
FullBath

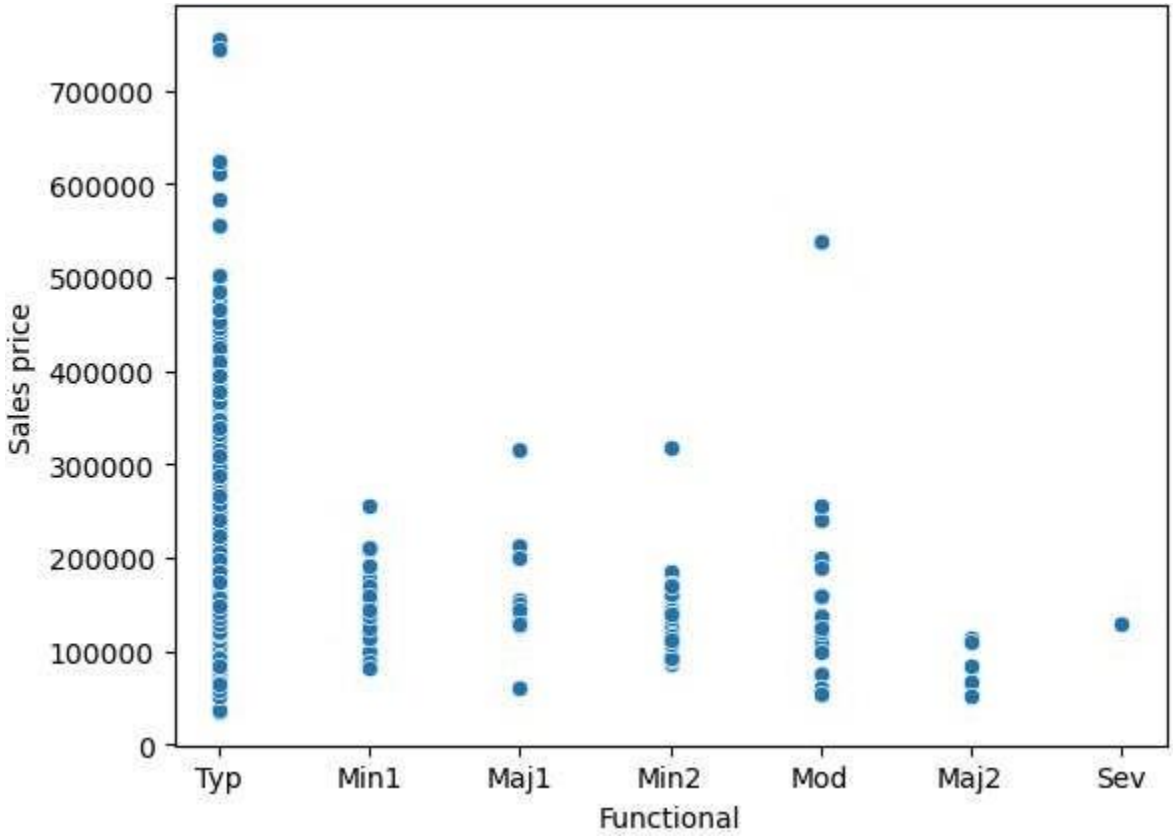

Sales price
700000
600000
500000
400000
300000
200000
100000
0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
HalfBath

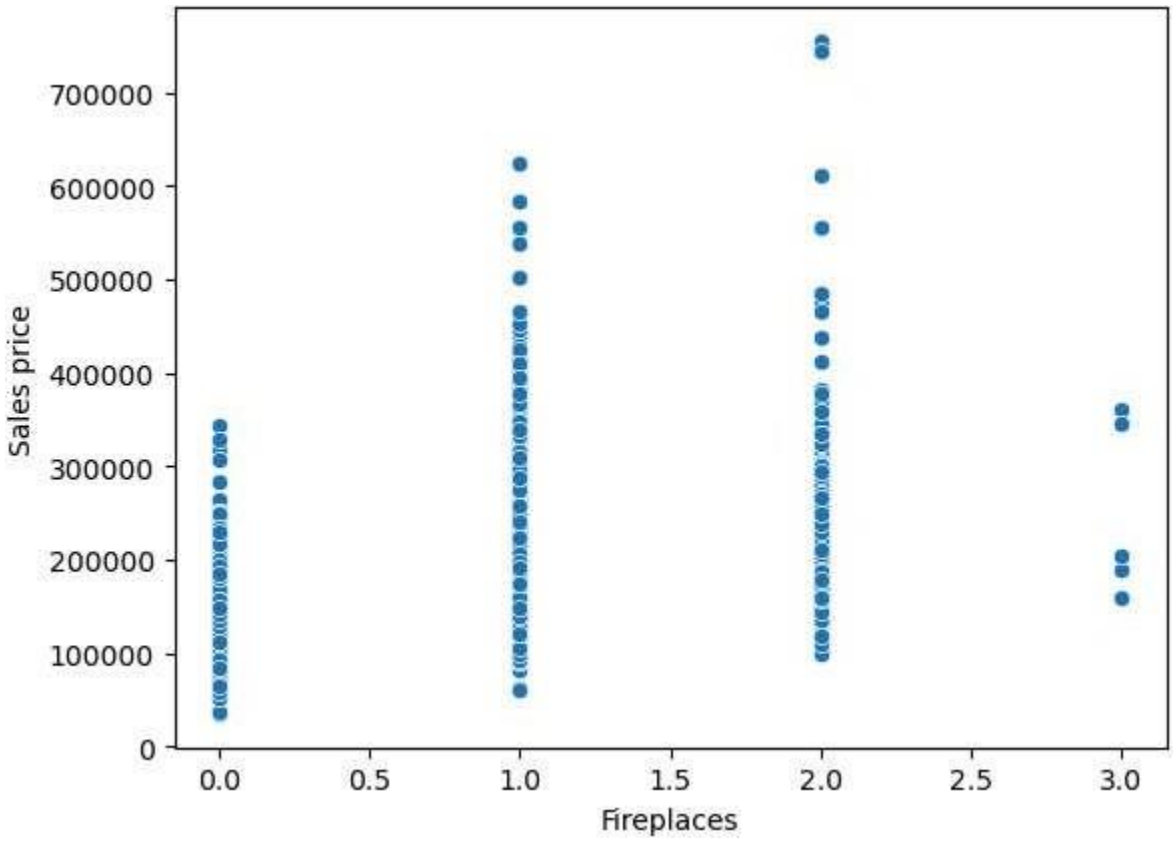

Sales price
700000
600000
500000
400000
300000
200000
100000
0
0
1
2
3
4
5
6
7
8
BedroomAbvGr

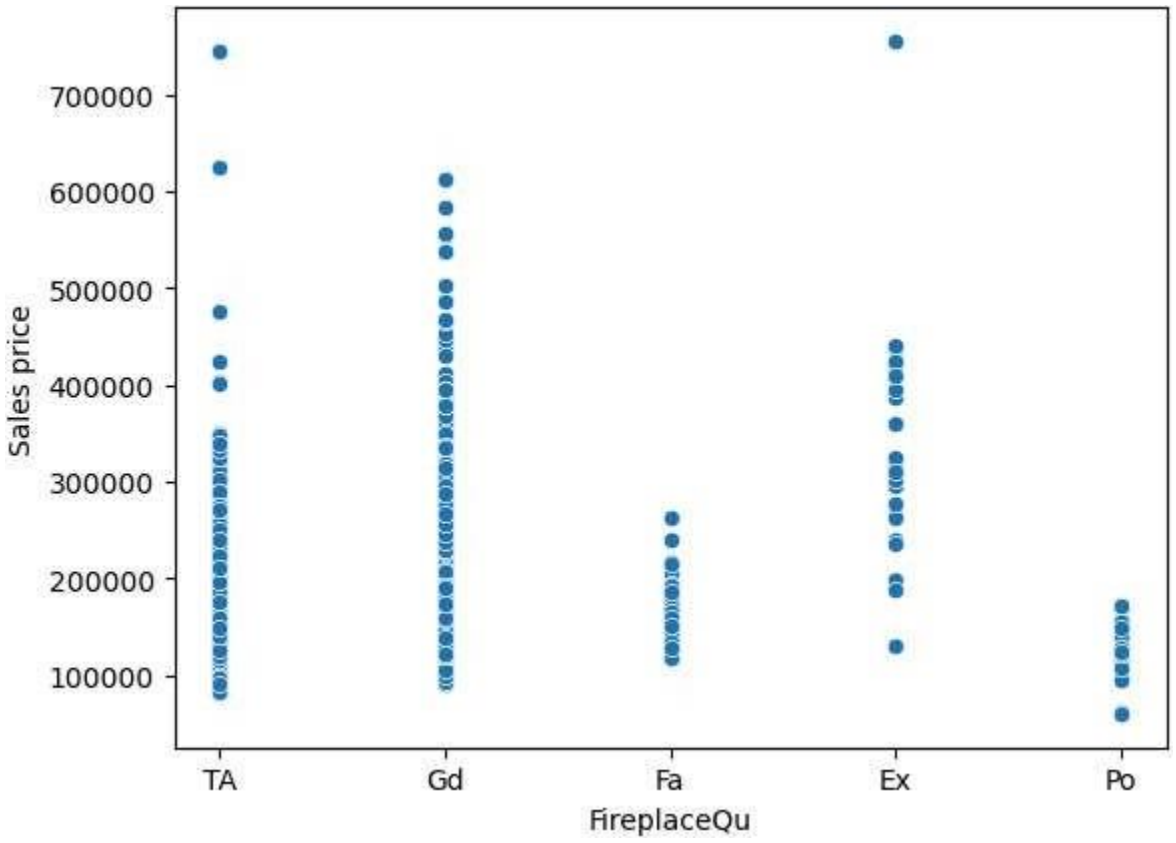

Sales price
700000
600000
500000
400000
300000
200000
100000
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
KitchenAbvGr

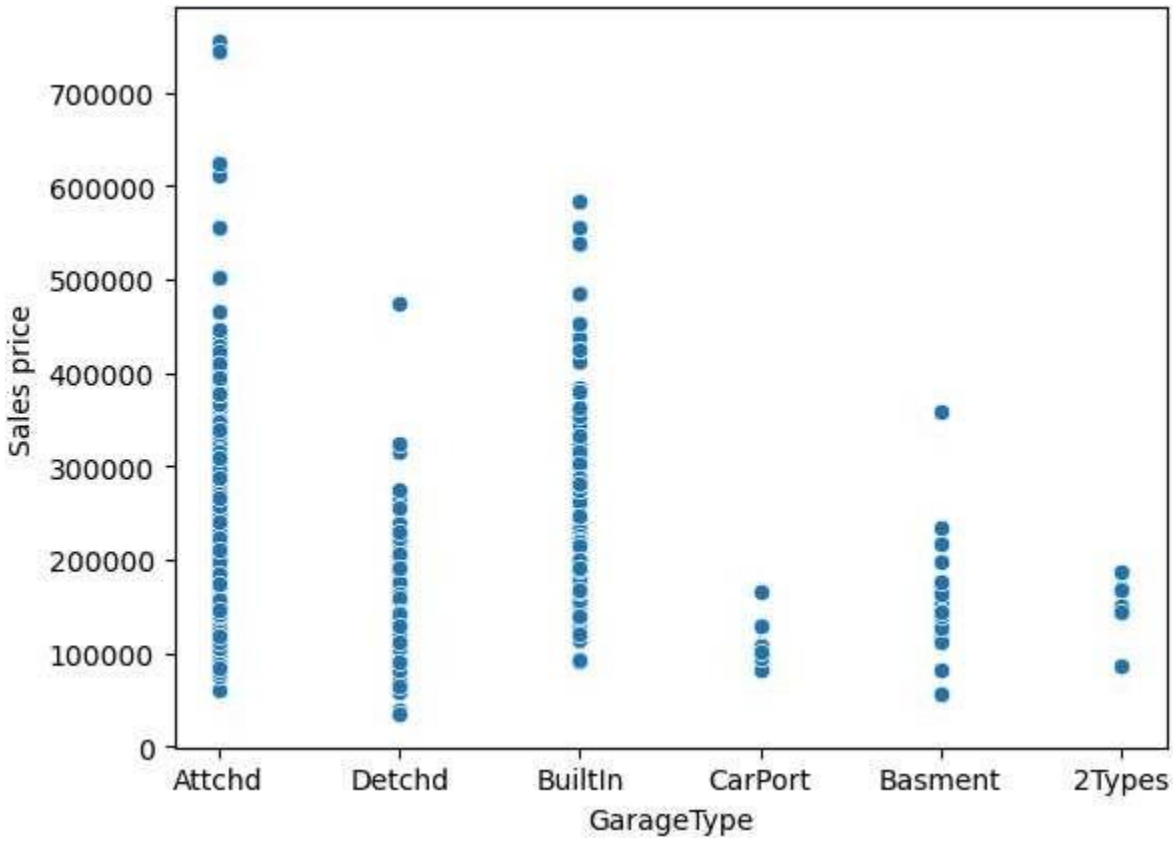

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Gd
TA
Ex
Fa
KitchenQual

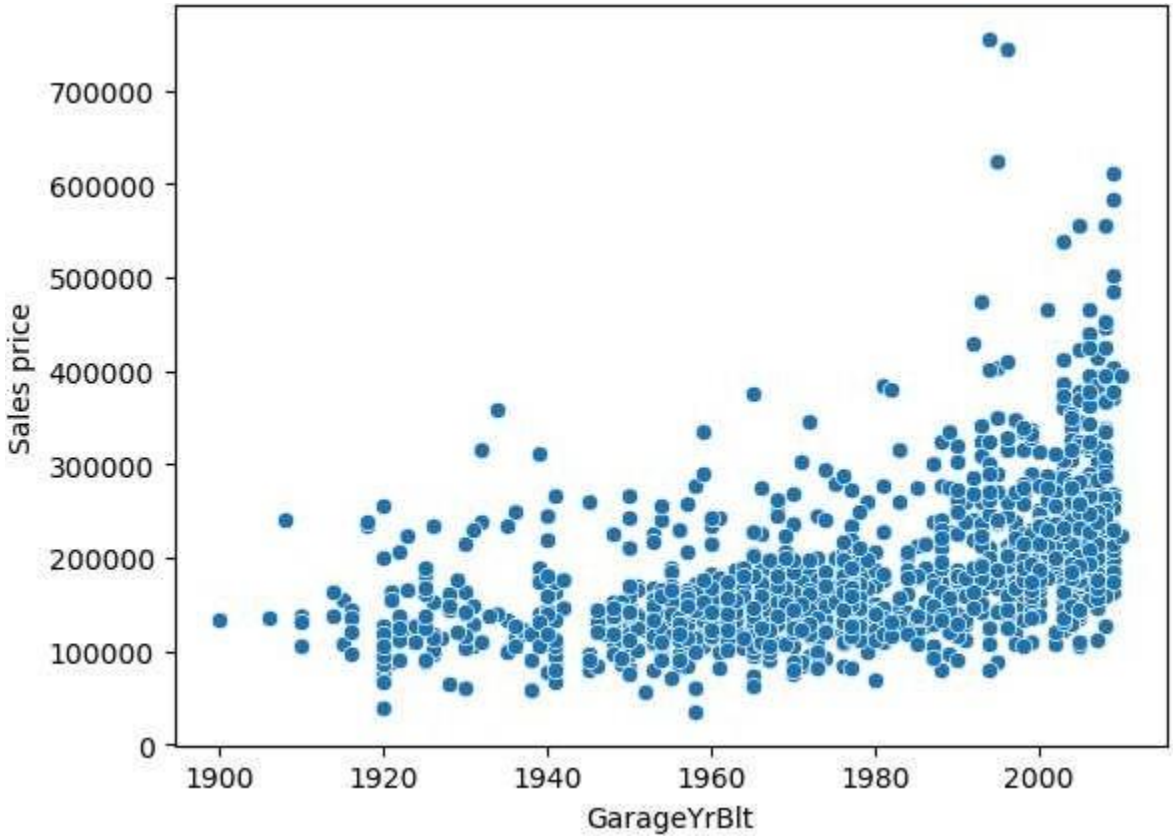

Sales price
700000
600000
500000
400000
300000
200000
100000
0
2
4
6
8
10
12
14
TotRmsAbvGrd

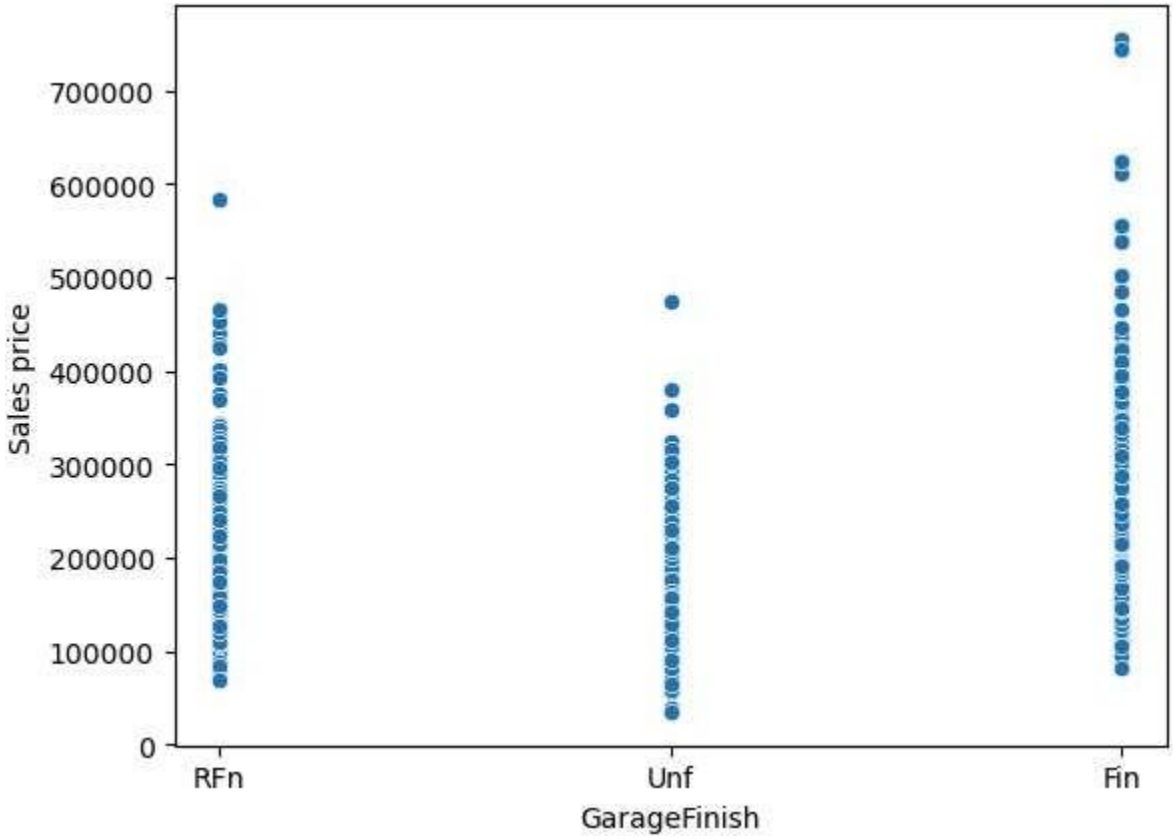

Sales price
700000
600000
500000
400000
300000
200000
100000
0
Typ
Min1
Maj1
Min2
Mod
Maj2
Sev
Functional

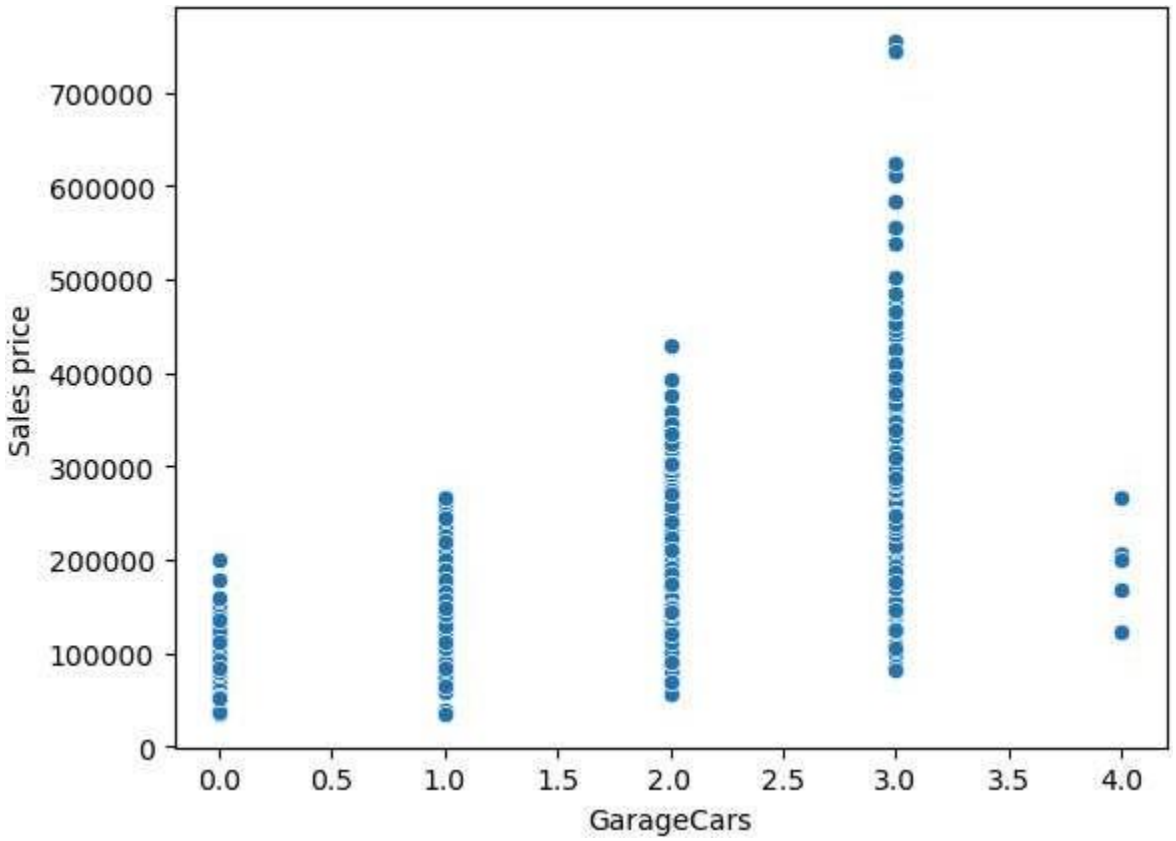

Sales price
700000
600000
500000
400000
300000
200000
100000
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
Fireplaces

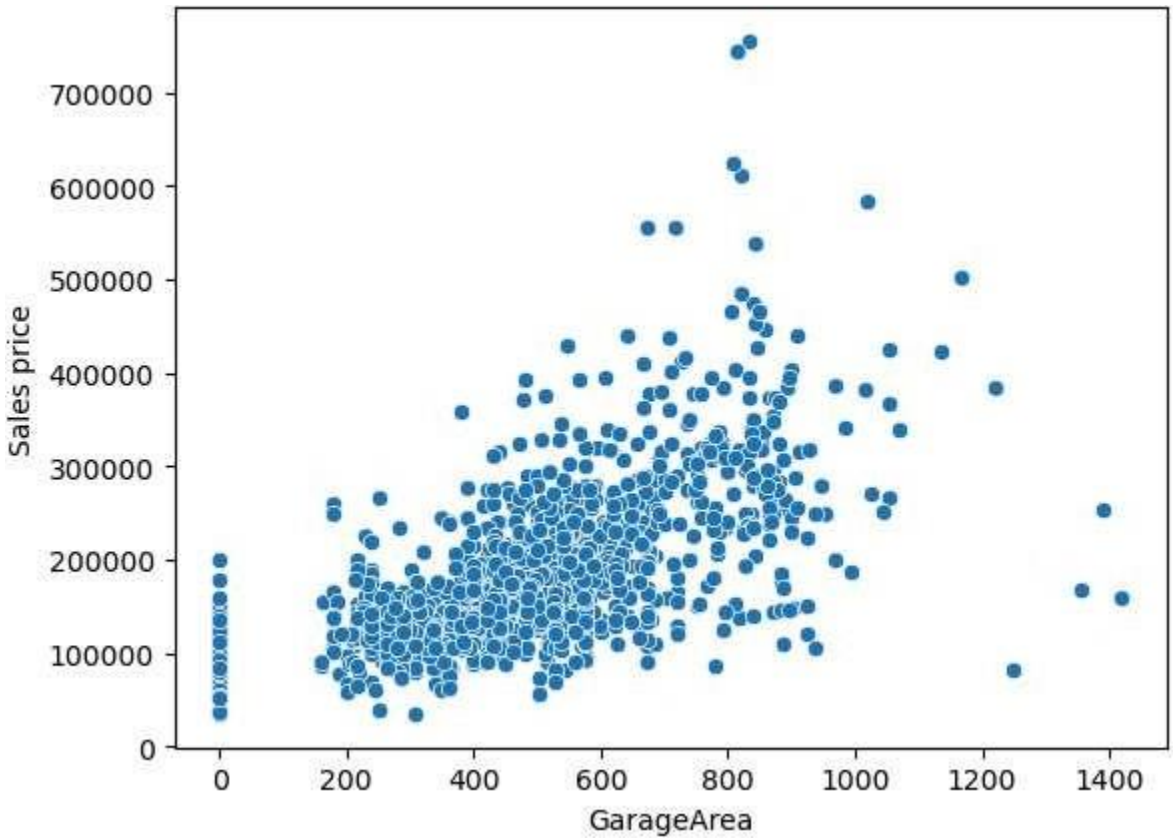

700000
600000
500000
400000
300000
200000
100000
Sales price
TA
Gd
Fa
Ex
Po
FireplaceQu

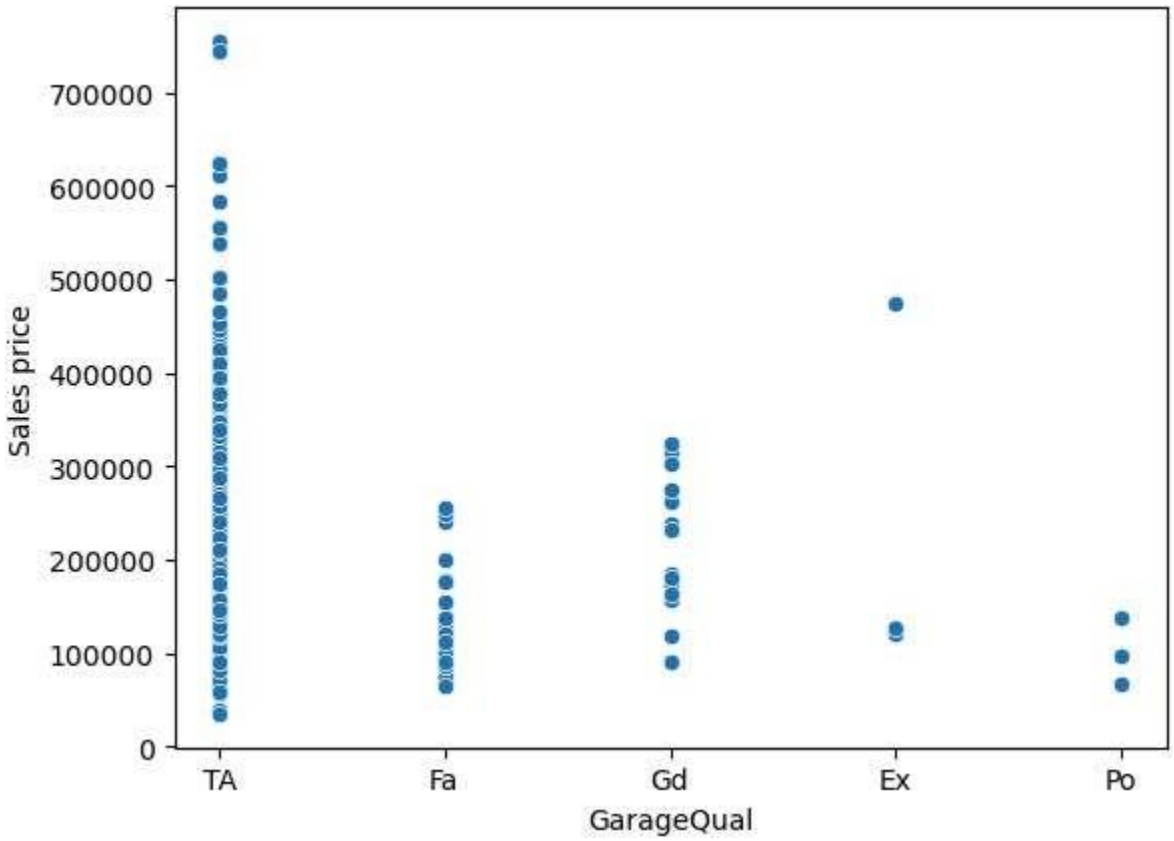

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Attchd
Detchd
BuiltIn
CarPort
Basment
2Types
GarageType

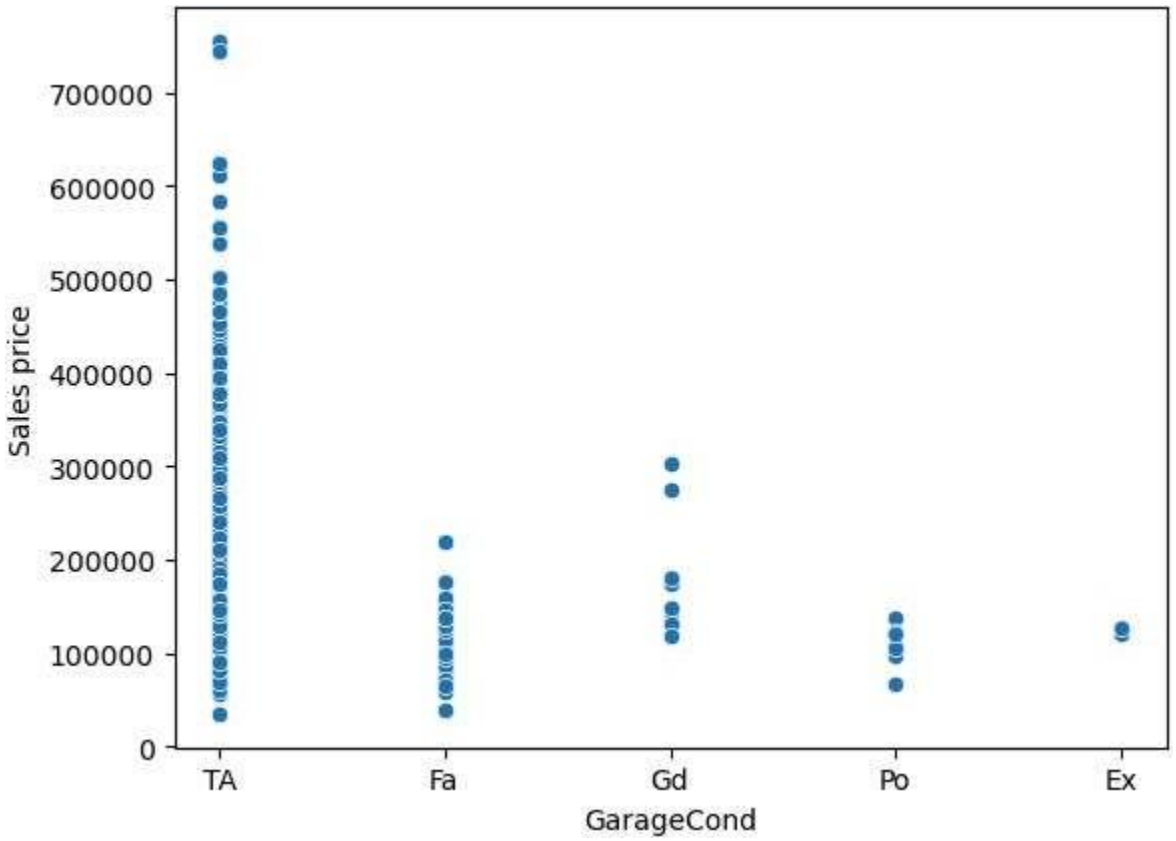

Sales price
700000
600000
500000
400000
300000
200000
100000
0
1900
1920
1940
1960
1980
2000
GarageYrBlt

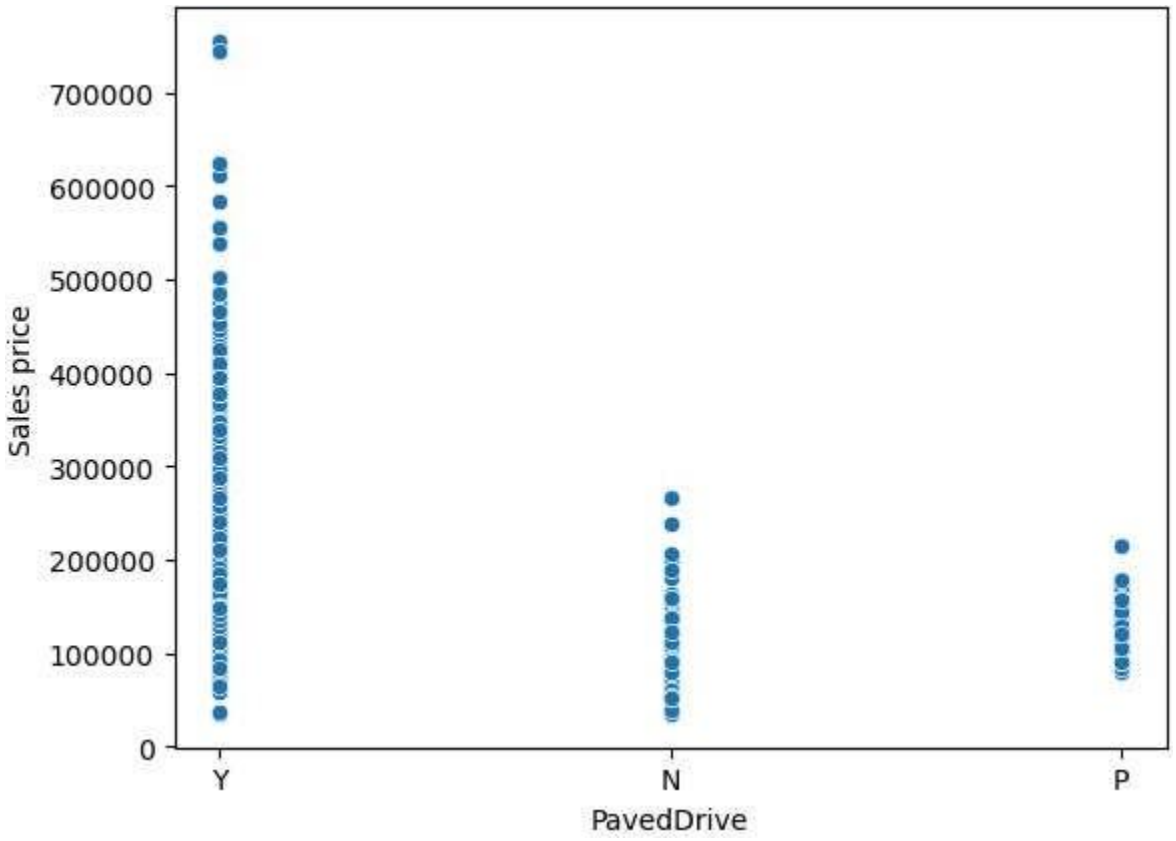

700000
600000
500000
400000
300000
200000
100000
0
Sales price
RFn
Unf
Fin
GarageFinish

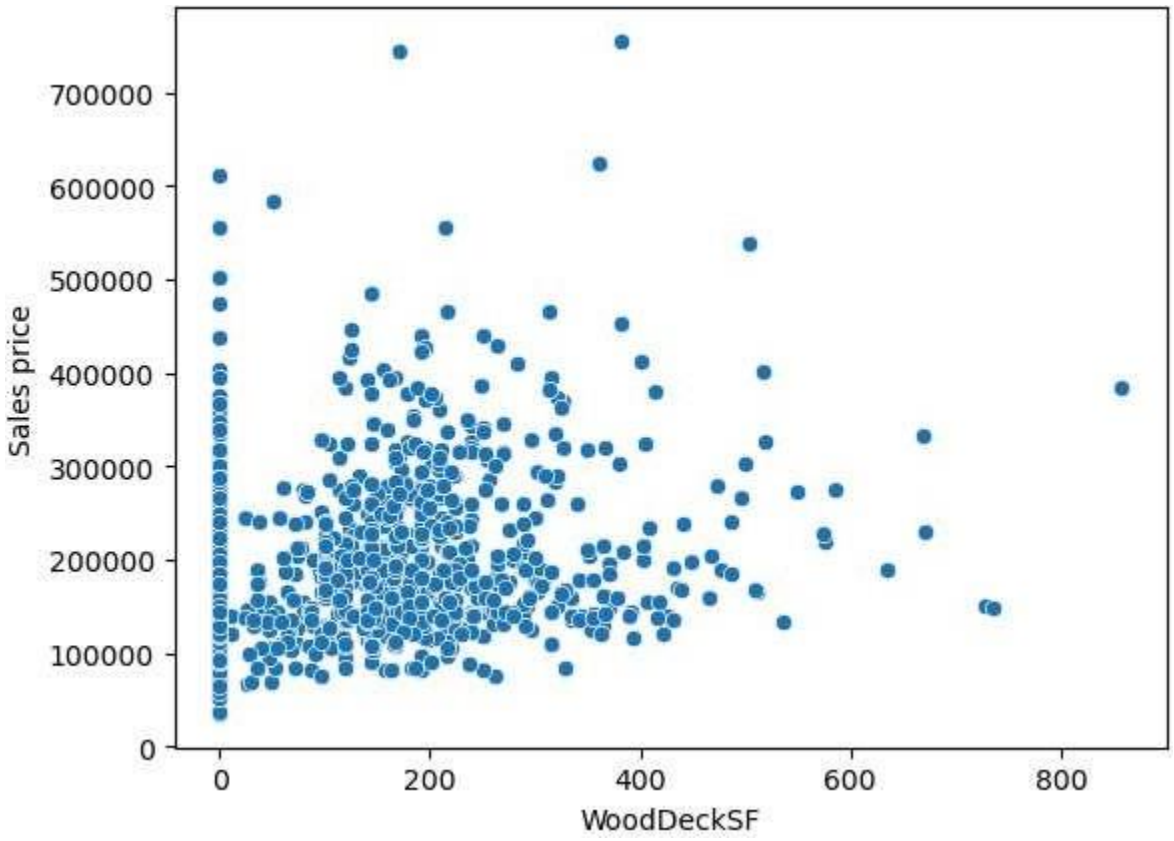

Sales price
700000
600000
500000
400000
300000
200000
100000
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
GarageCars

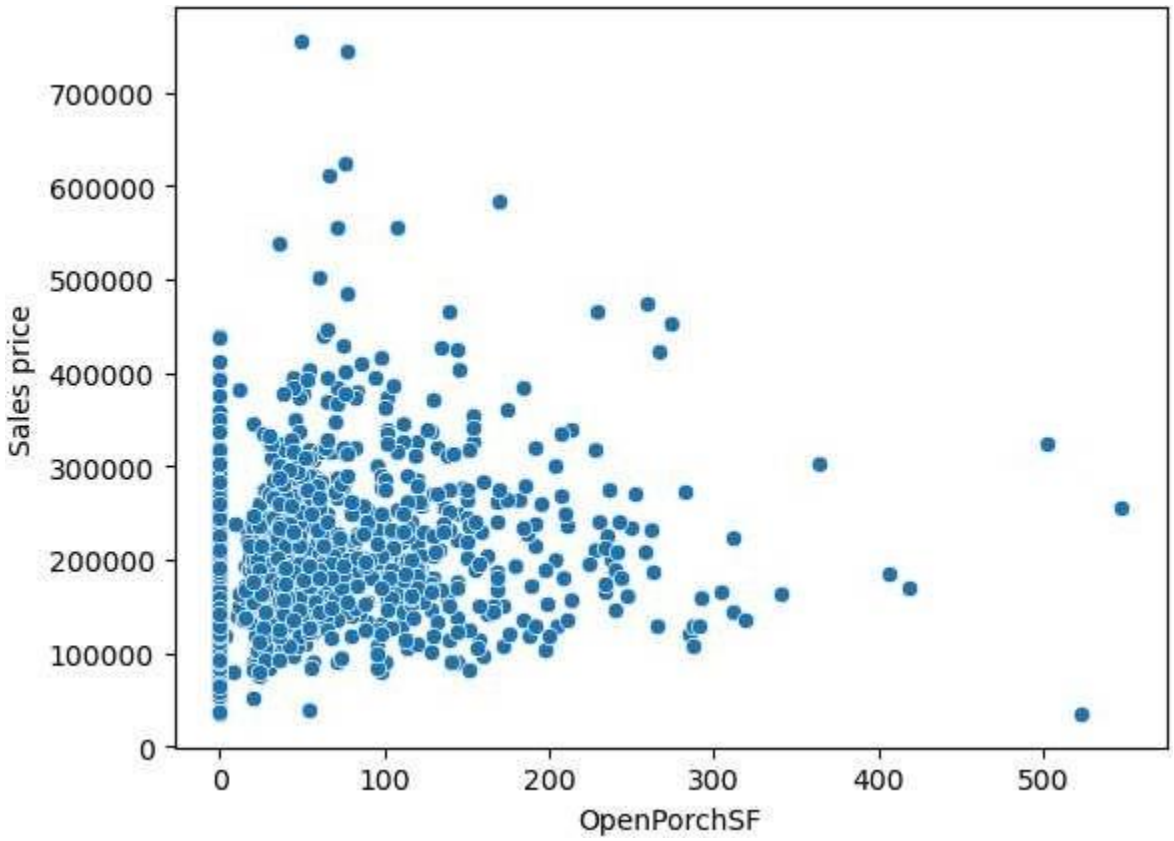

Sales price
GarageArea
0
100000
200000
300000
400000
500000
600000
700000
0
200
400
600
800
1000
1200
1400

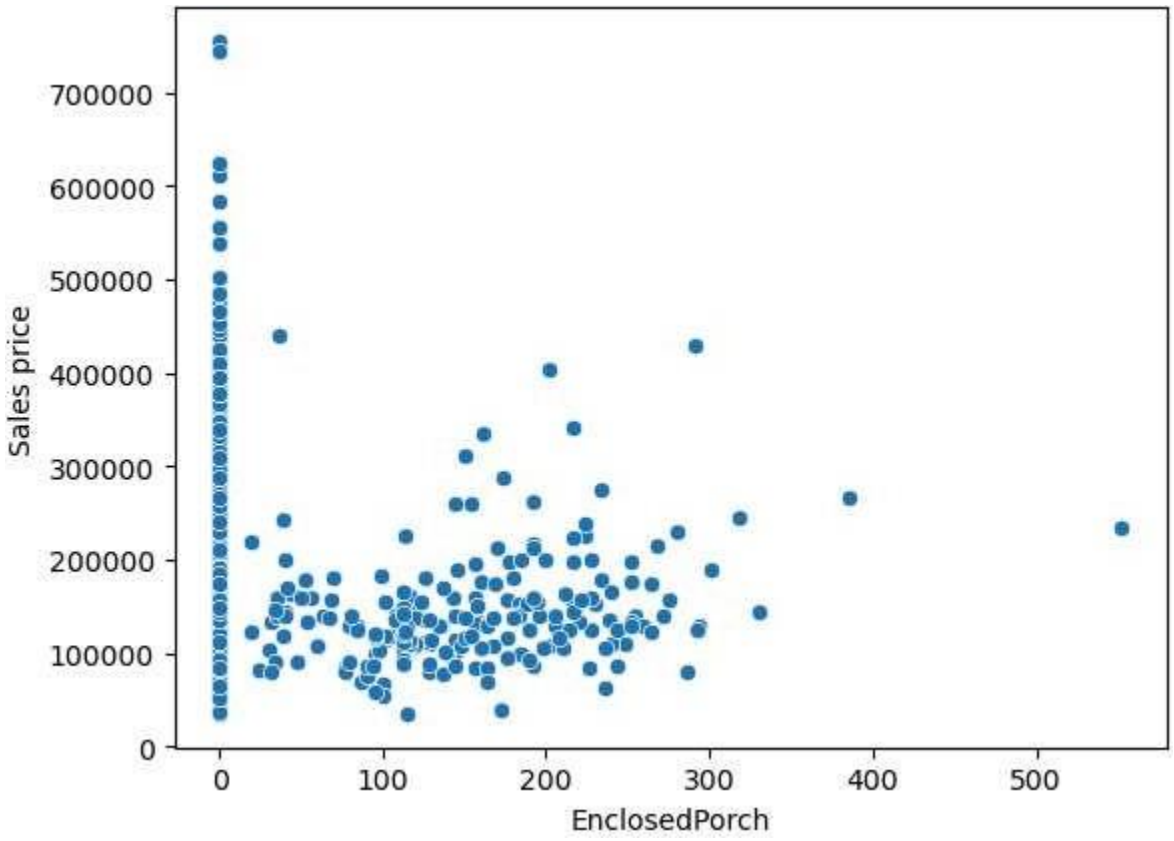

Sales price
700000
600000
500000
400000
300000
200000
100000
0
TA
Fa
Gd
Ex
Po
GarageQual

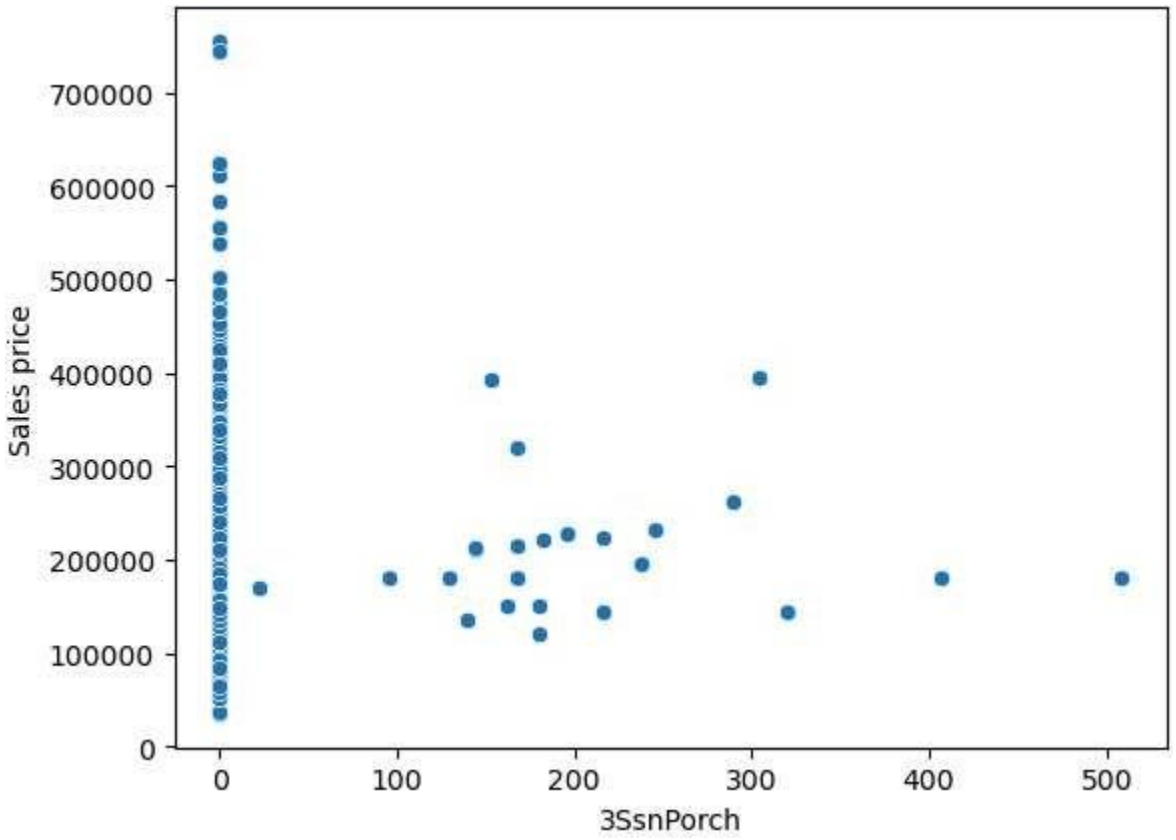

Sales price
700000
600000
500000
400000
300000
200000
100000
0
TA
Fa
Gd
Po
Ex
GarageCond

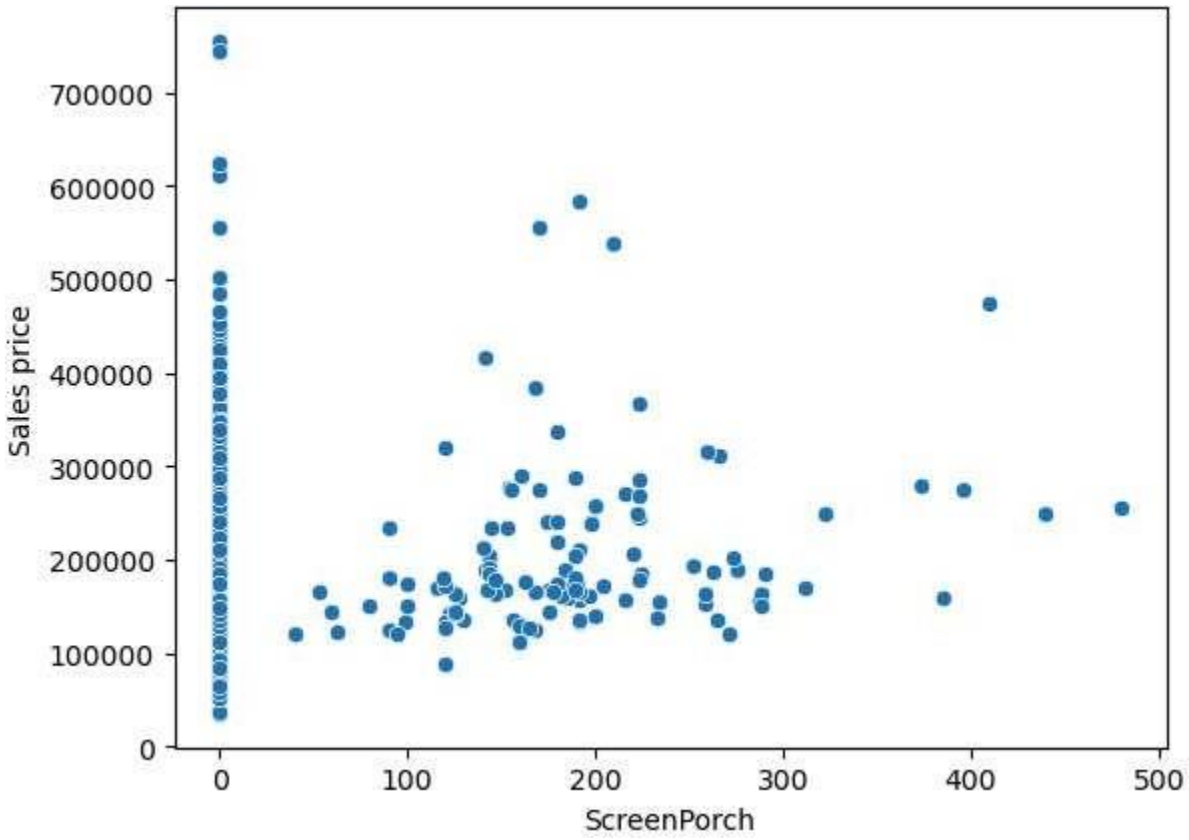

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Y
N
P
PavedDrive

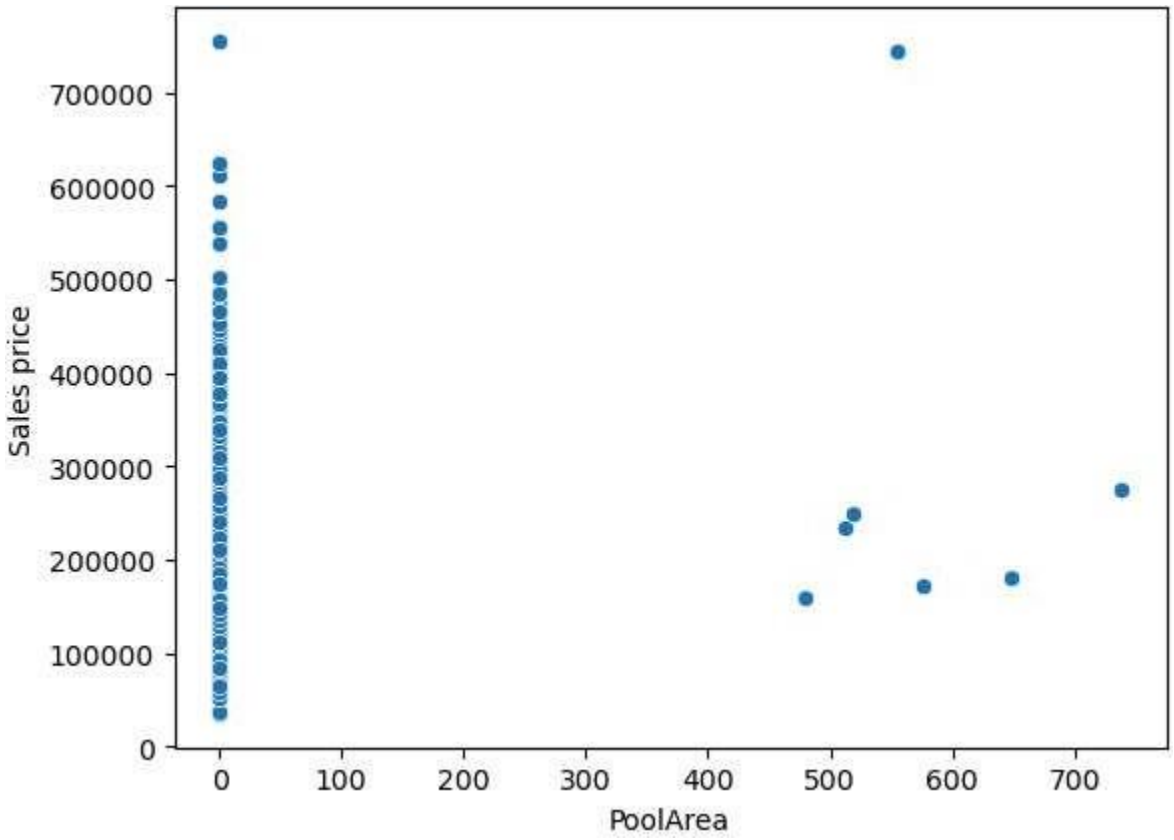

Sales price
700000
600000
500000
400000
300000
200000
100000
0
0
200
400
600
800
WoodDeckSF

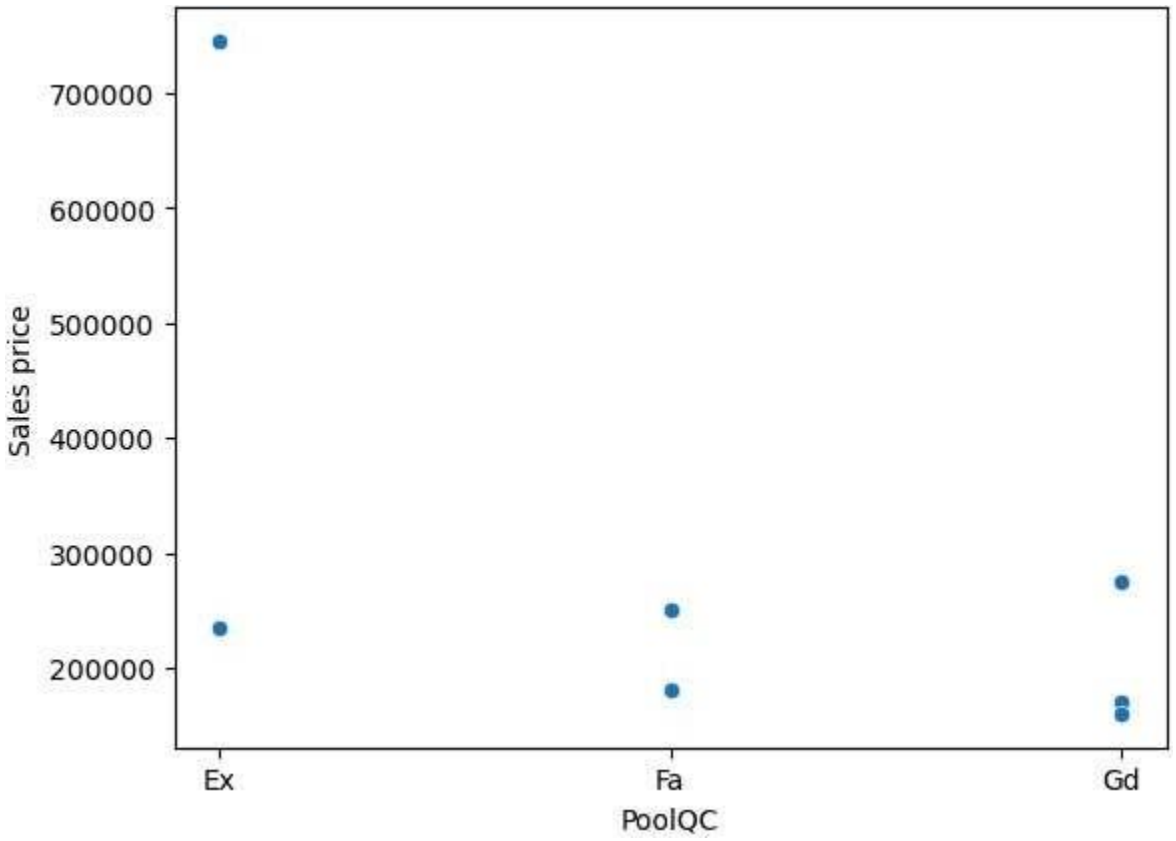

Sales price
700000
600000
500000
400000
300000
200000
100000
0
0
100
200
300
400
500
OpenPorchSF

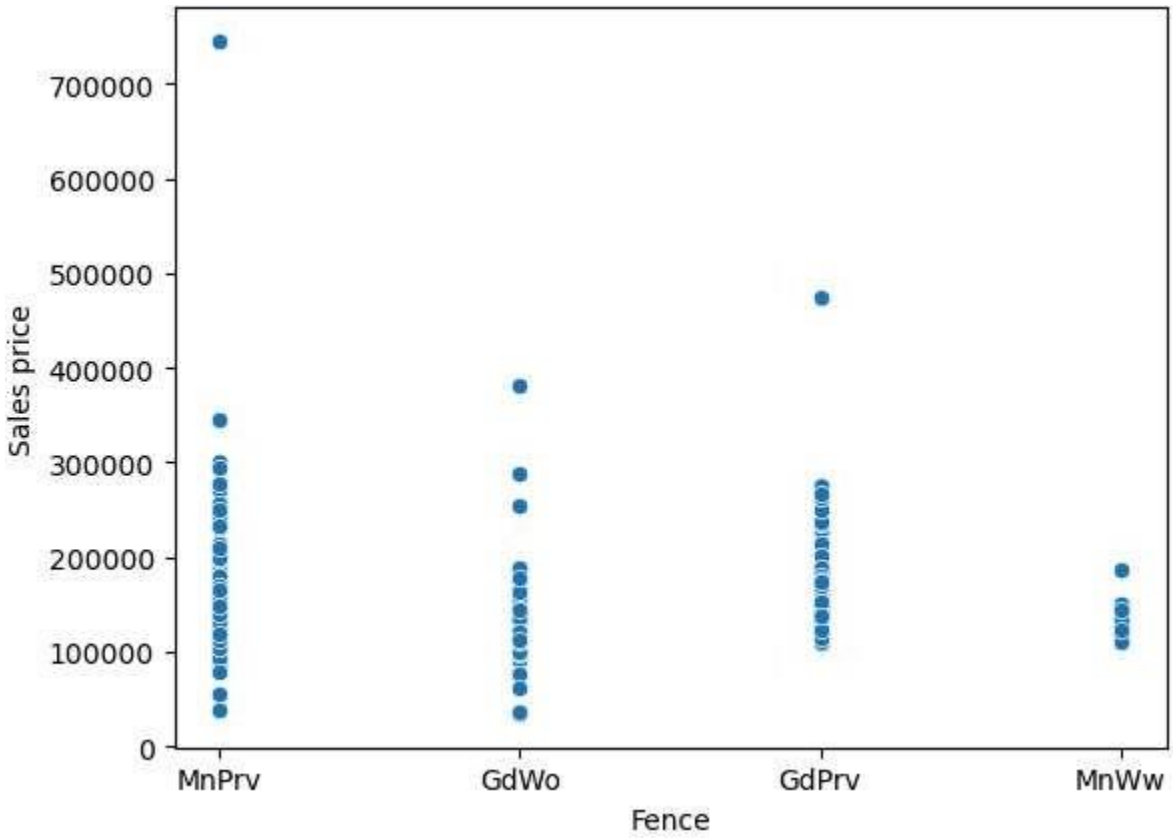

Sales price
700000
600000
500000
400000
300000
200000
100000
0
0
100
200
300
400
500
EnclosedPorch

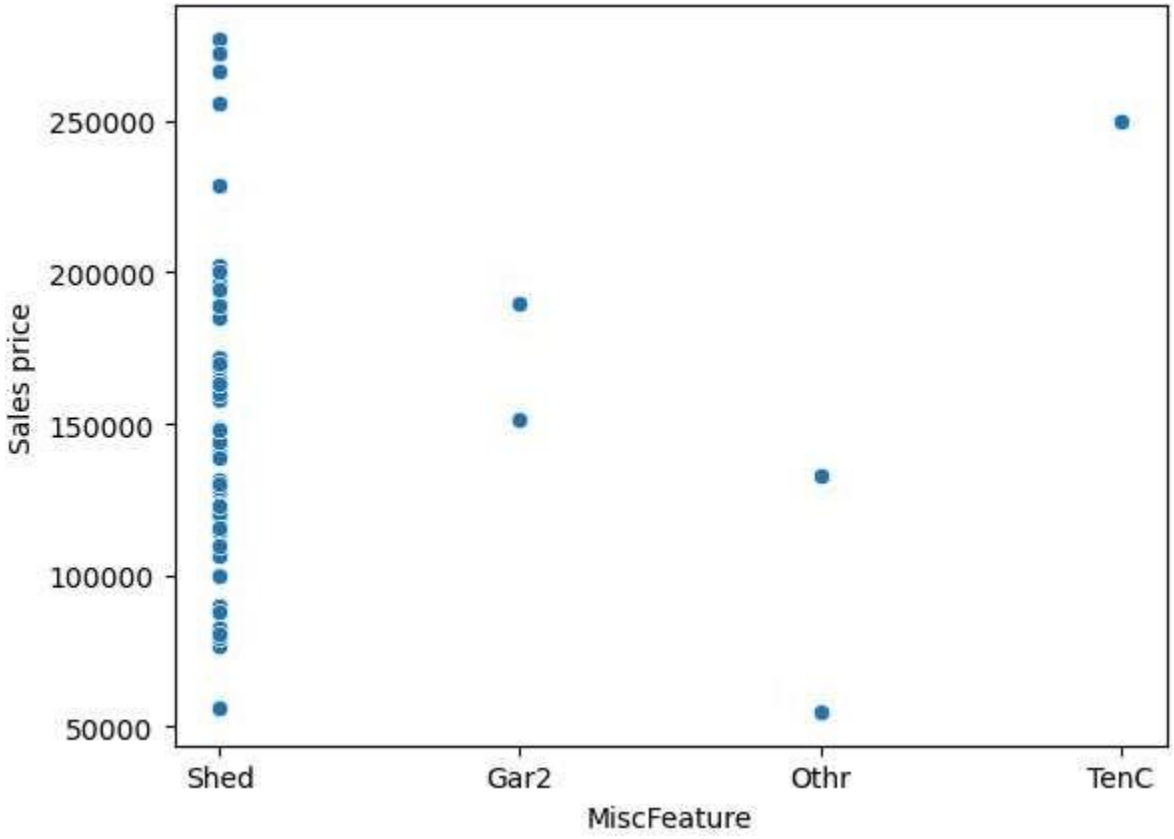

Sales price
700000
600000
500000
400000
300000
200000
100000
0
0
100
200
300
400
500
3SsnPorch

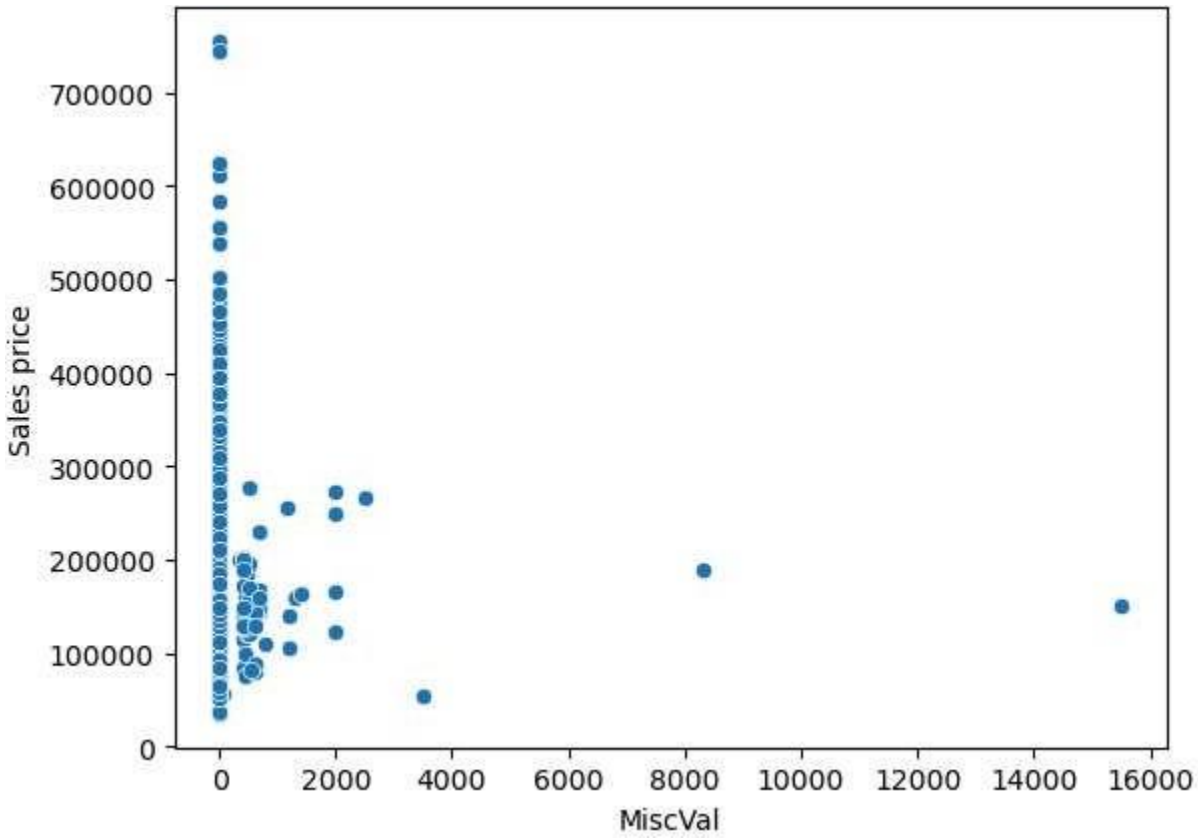

Sales price
ScreenPorch
0
100000
200000
300000
400000
500000
600000
700000
0
100
200
300
400
500

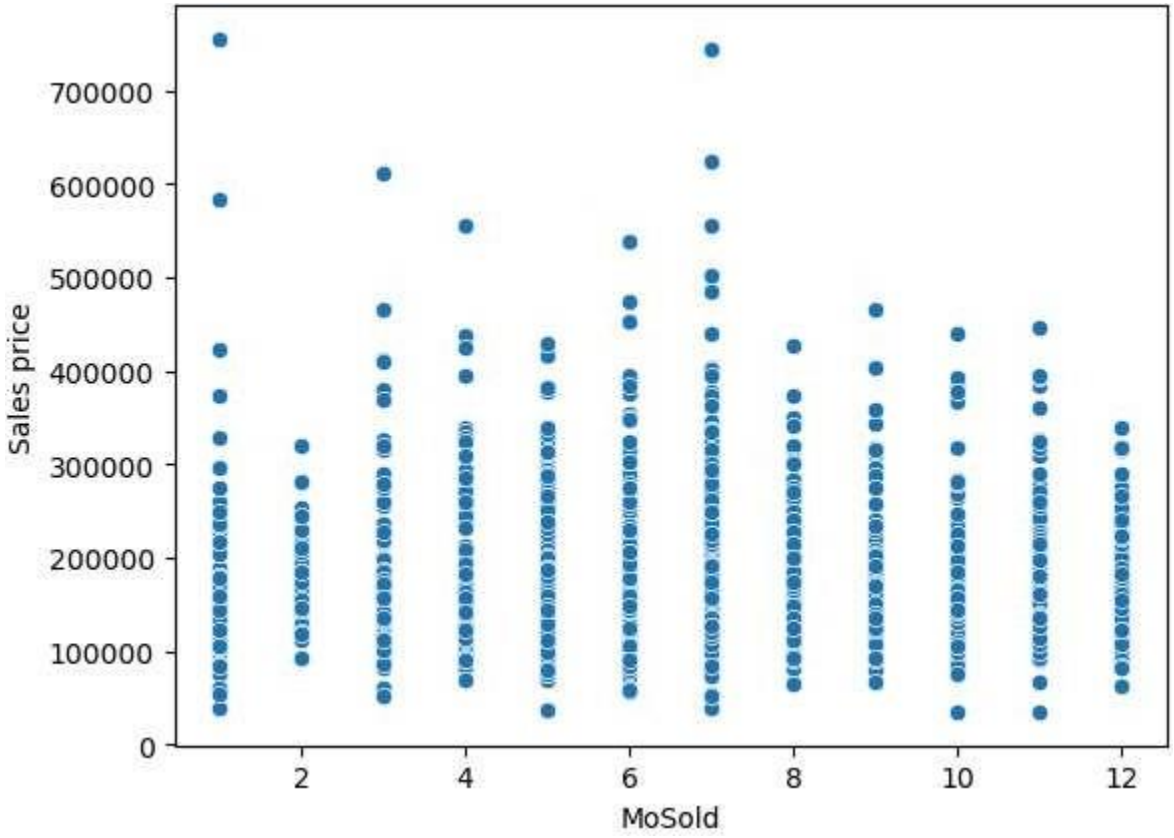

Sales price
PoolArea
0
100000
200000
300000
400000
500000
600000
700000
0
100
200
300
400
500
600
700

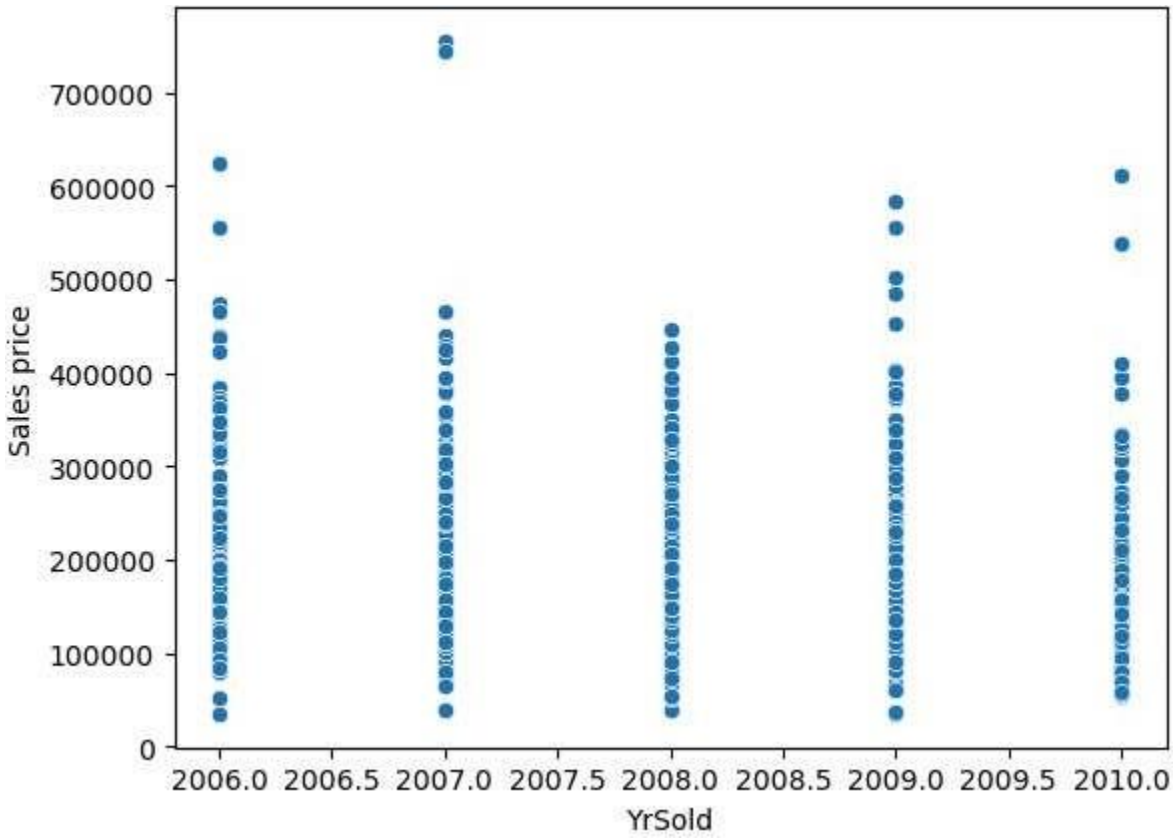

700000
600000
500000
400000
300000
200000
Sales price
Ex
Fa
Gd
PoolQC

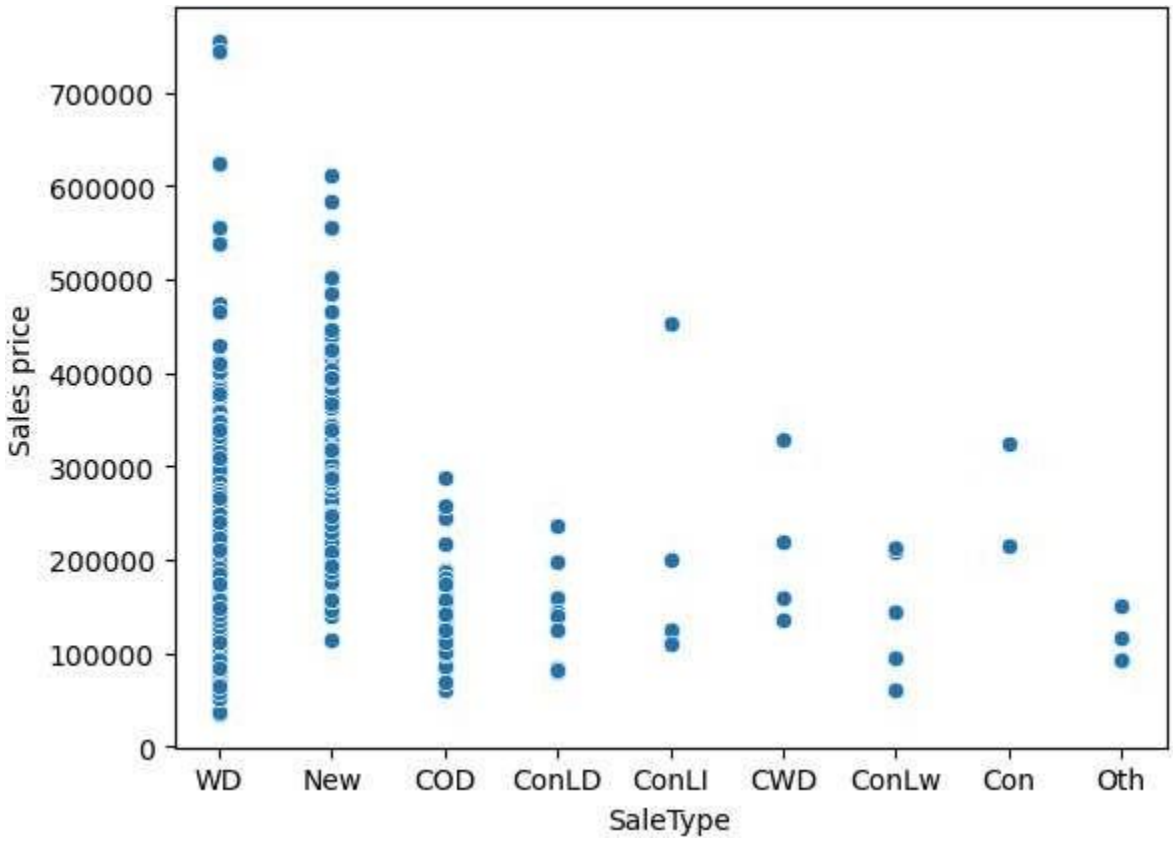

700000
600000
500000
400000
300000
200000
100000
0
Sales price
MnPrv
GdWo
GdPrv
MnWw
Fence

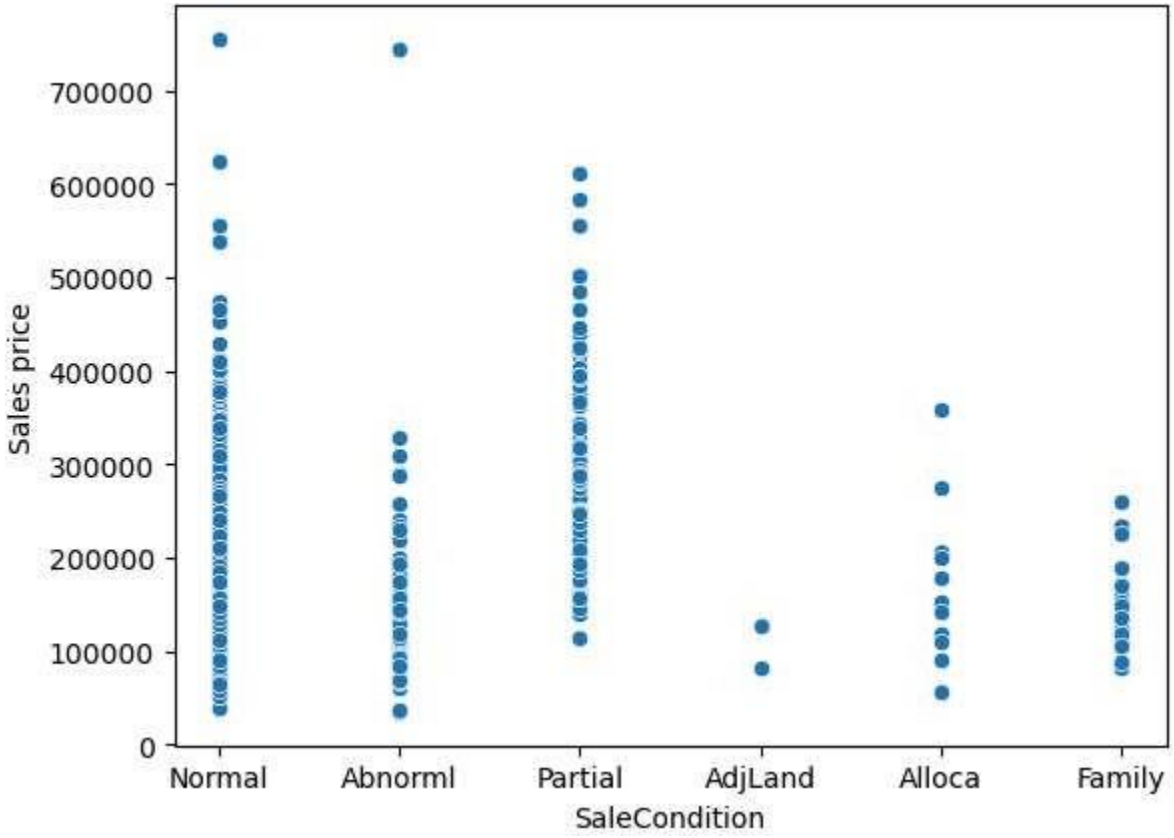

250000
200000
150000
100000
50000
Sales price
Shed
Gar2
Othr
TenC
MiscFeature


Sales price
MiscVal
0
100000
200000
300000
400000
500000
600000
700000
0
2000
4000
6000
8000
10000
12000
14000
16000


Sales price
MoSold
0
100000
200000
300000
400000
500000
600000
700000
2
4
6
8
10
12


Sales price
700000
600000
500000
400000
300000
200000
100000
0
2006.0
2006.5
2007.0
2007.5
2008.0
2008.5
2009.0
2009.5
2010.0
YrSold

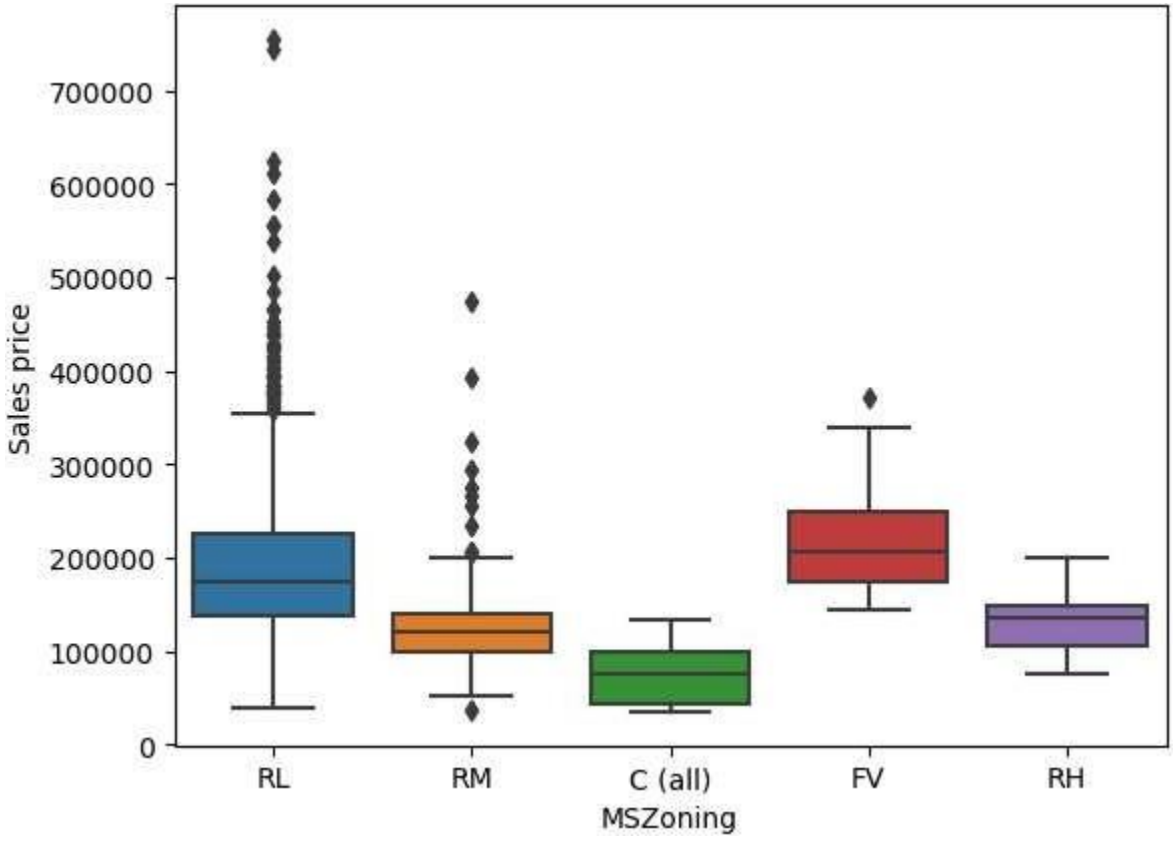

Sales price
700000
600000
500000
400000
300000
200000
100000
0
WD
New
COD
ConLD
ConLI
CWD
ConLw
Con
Oth
SaleType

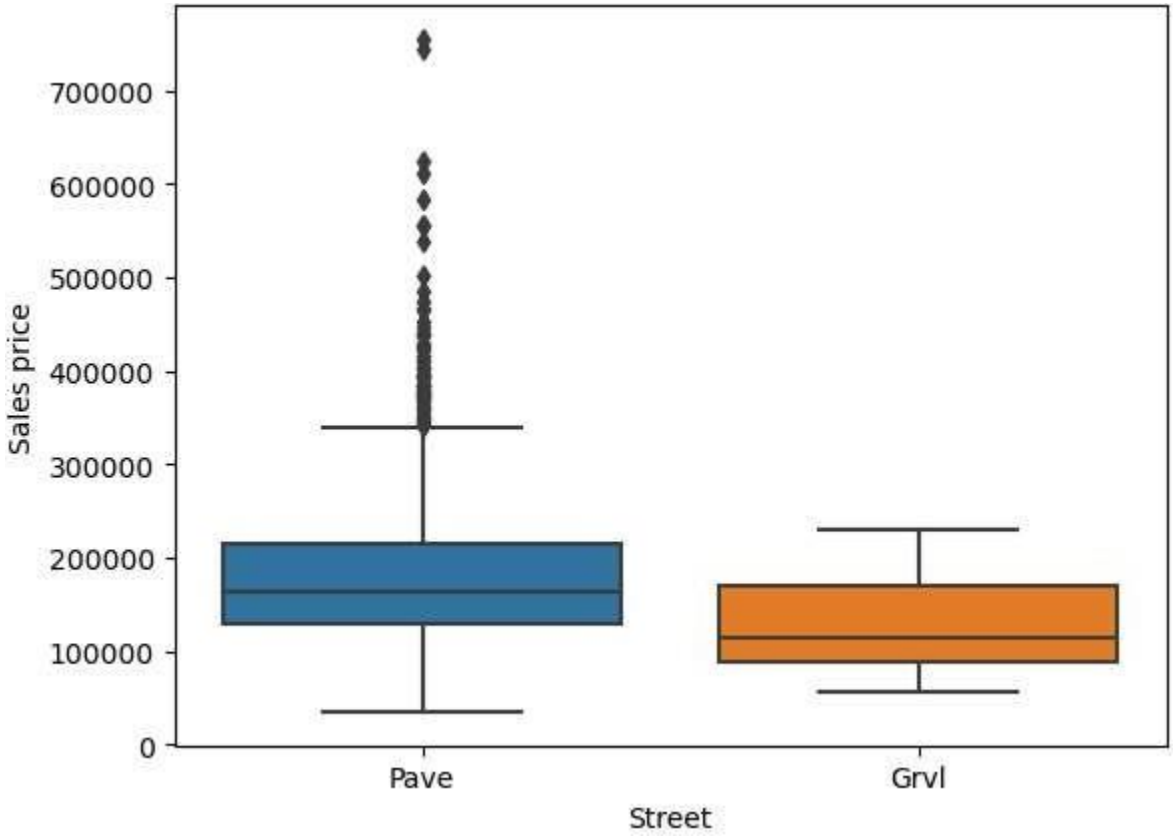


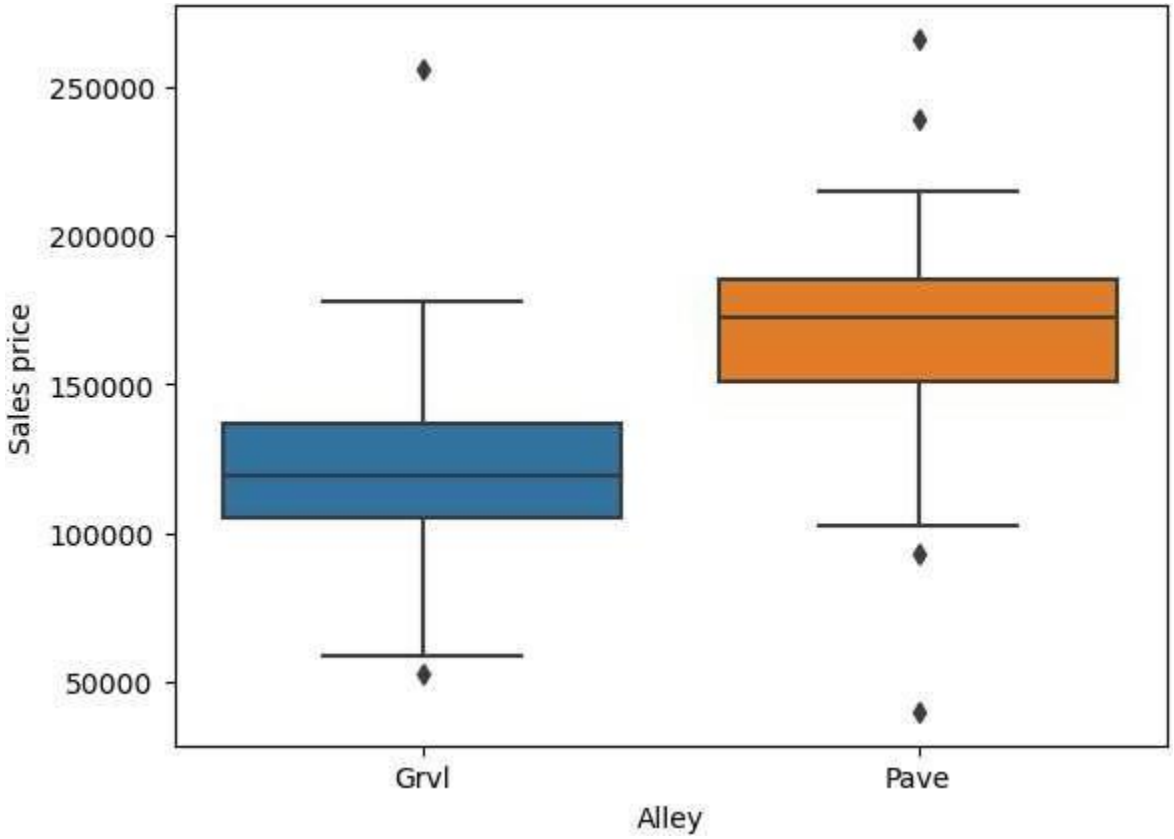
Note : Here clearly visible that there is a relation between the sale price vs contineous features

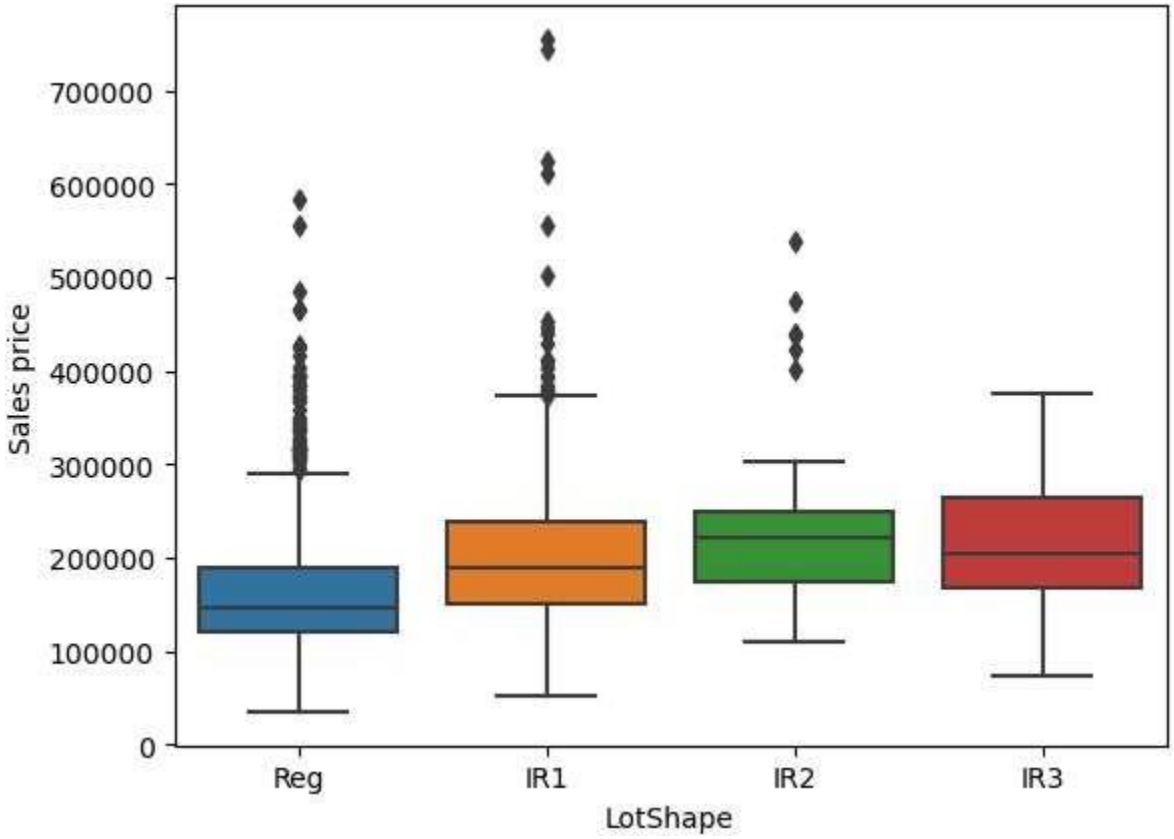
cat vs con(Saleprice)

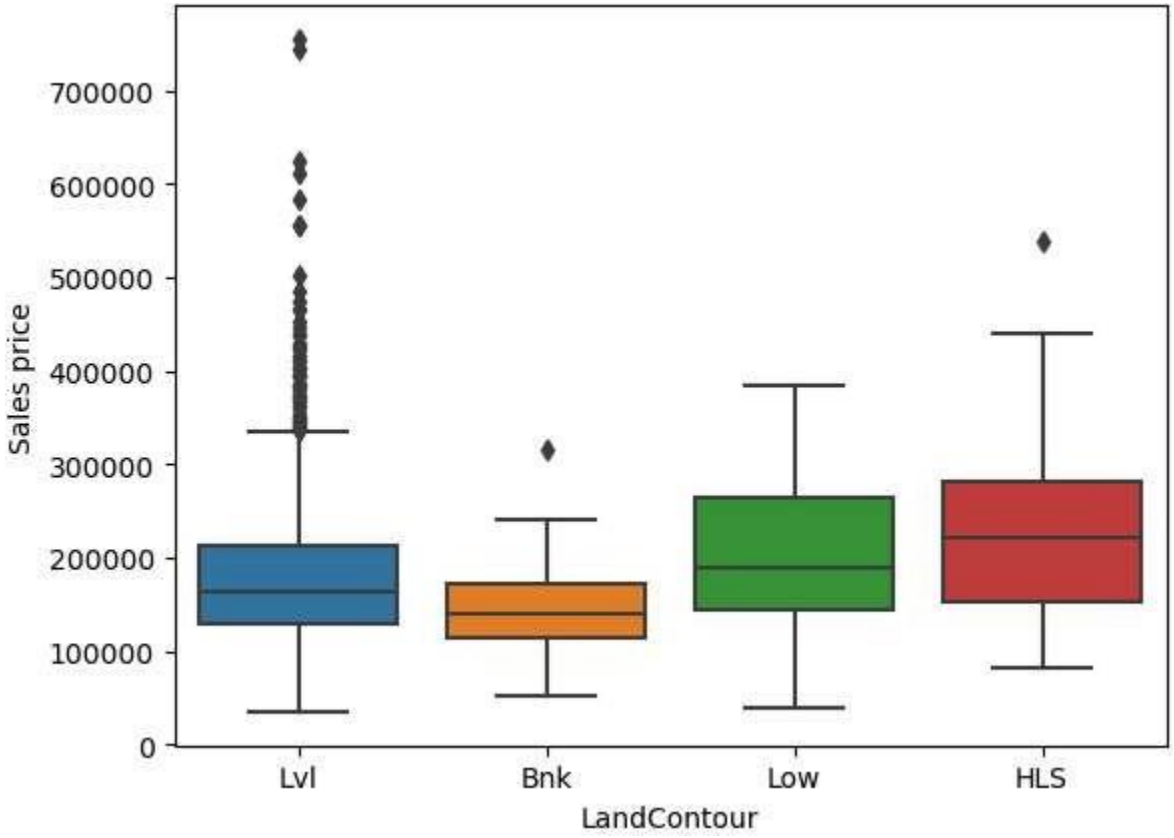
In [13]:
```
for i in df.columns[df.dtypes=='object']:
    sns.boxplot(data=df,x=i, y='SalePrice')
    plt.xlabel(i)
    plt.ylabel('Sales price')
    plt.show()
```

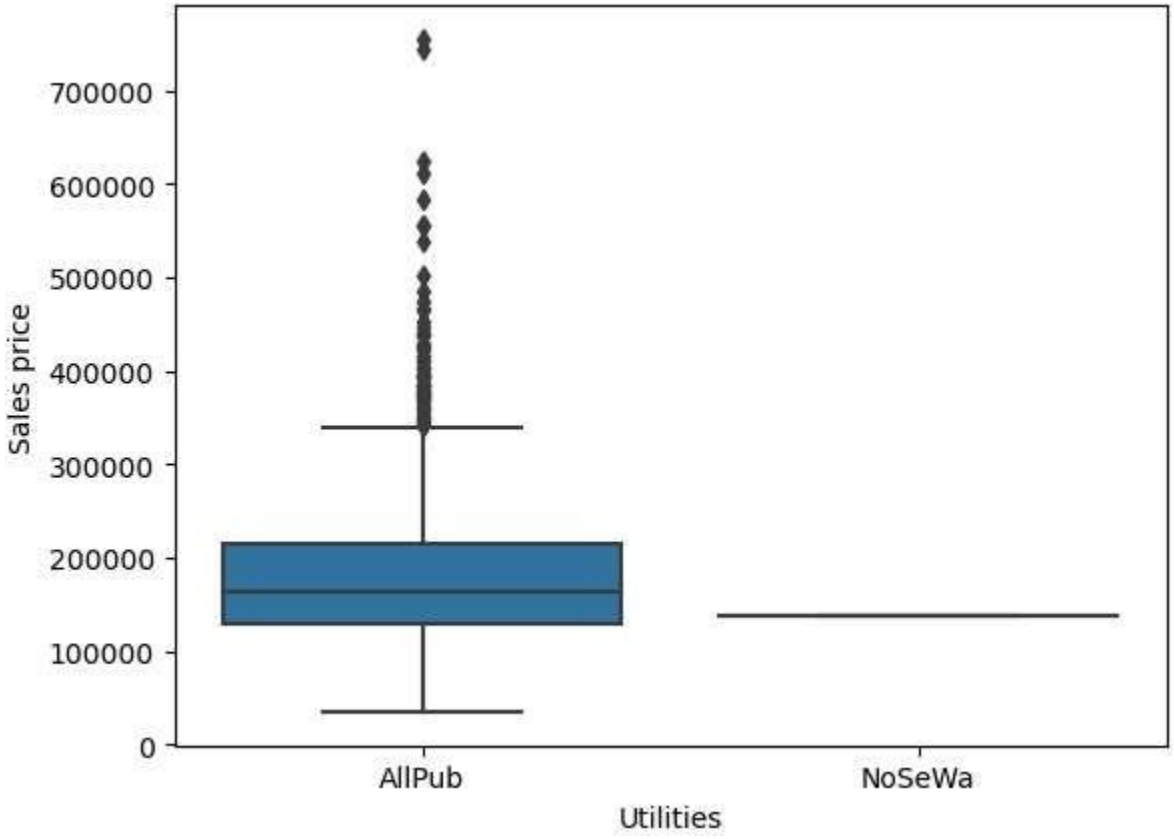

700000
600000
500000
400000
300000
200000
100000
0
Sales price
RL
RM
C (all)
FV
RH
MSZoning

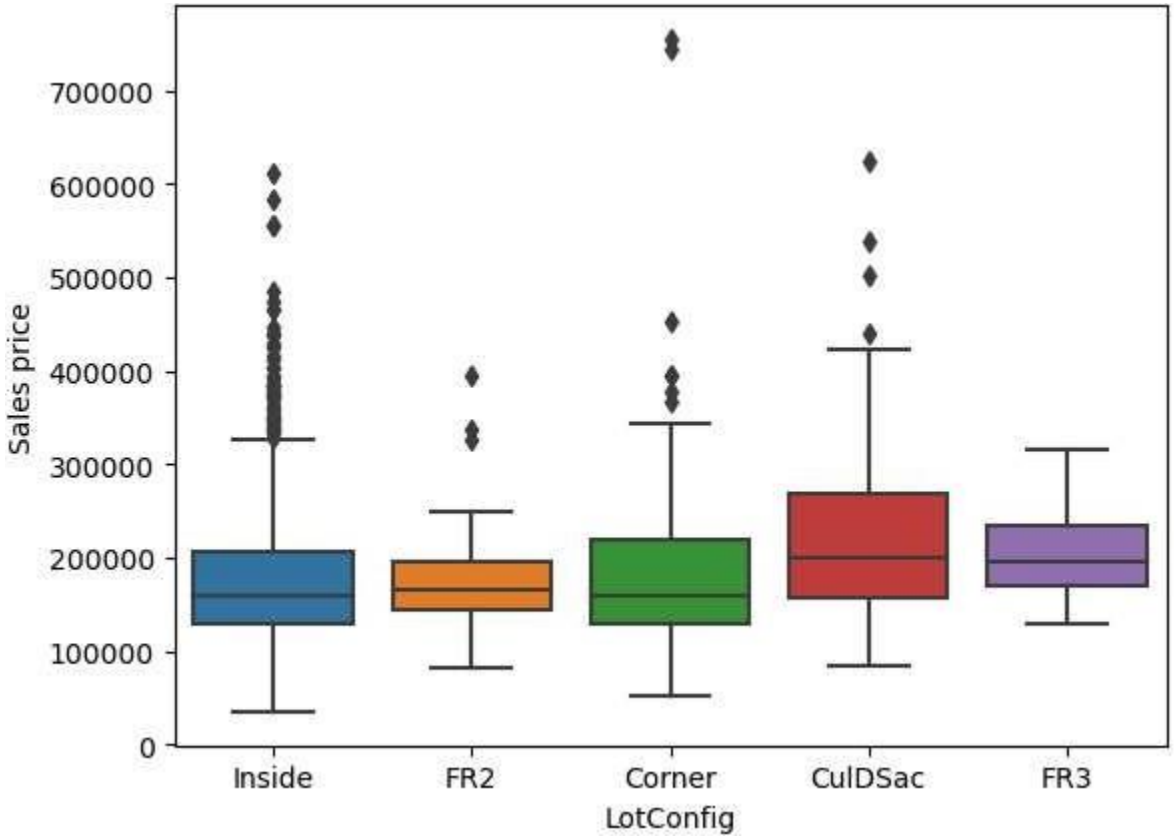

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Pave
Grvl
Street

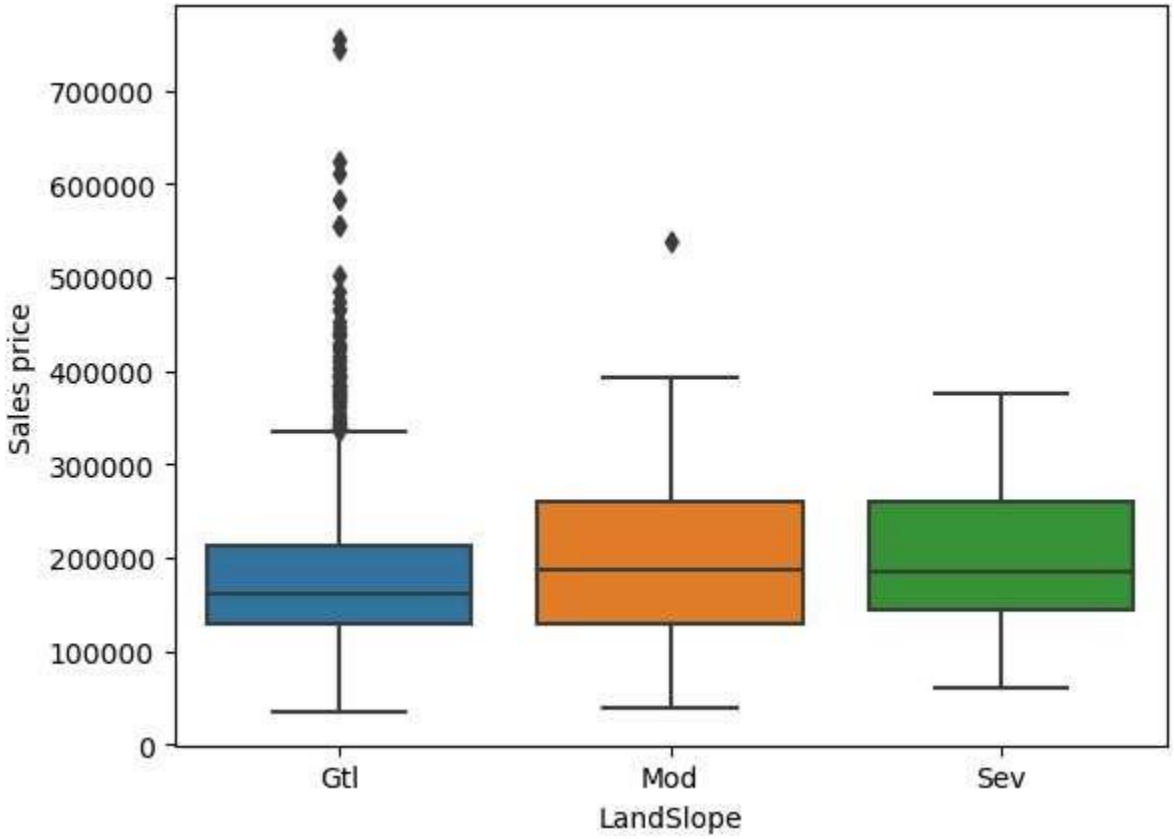

250000
200000
150000
100000
50000
Sales price
Grvl
Pave
Alley

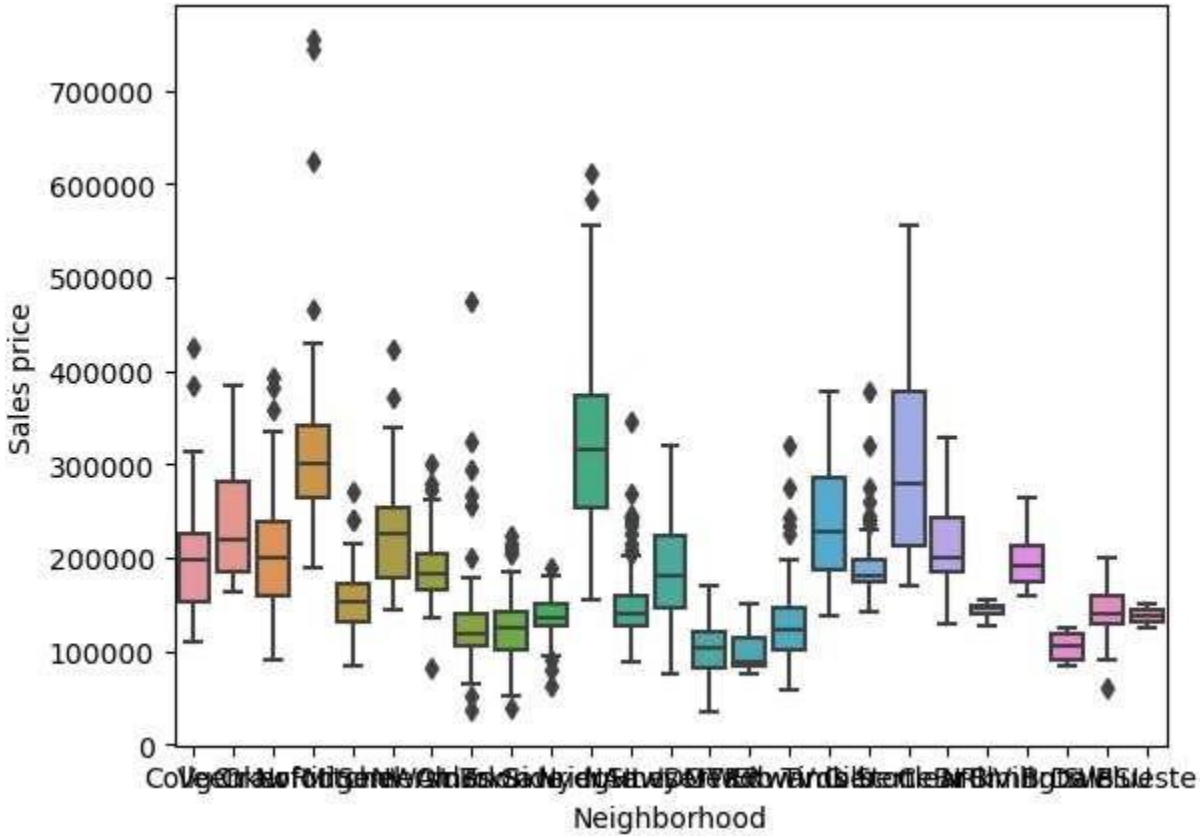

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Reg
IR1
IR2
IR3
LotShape

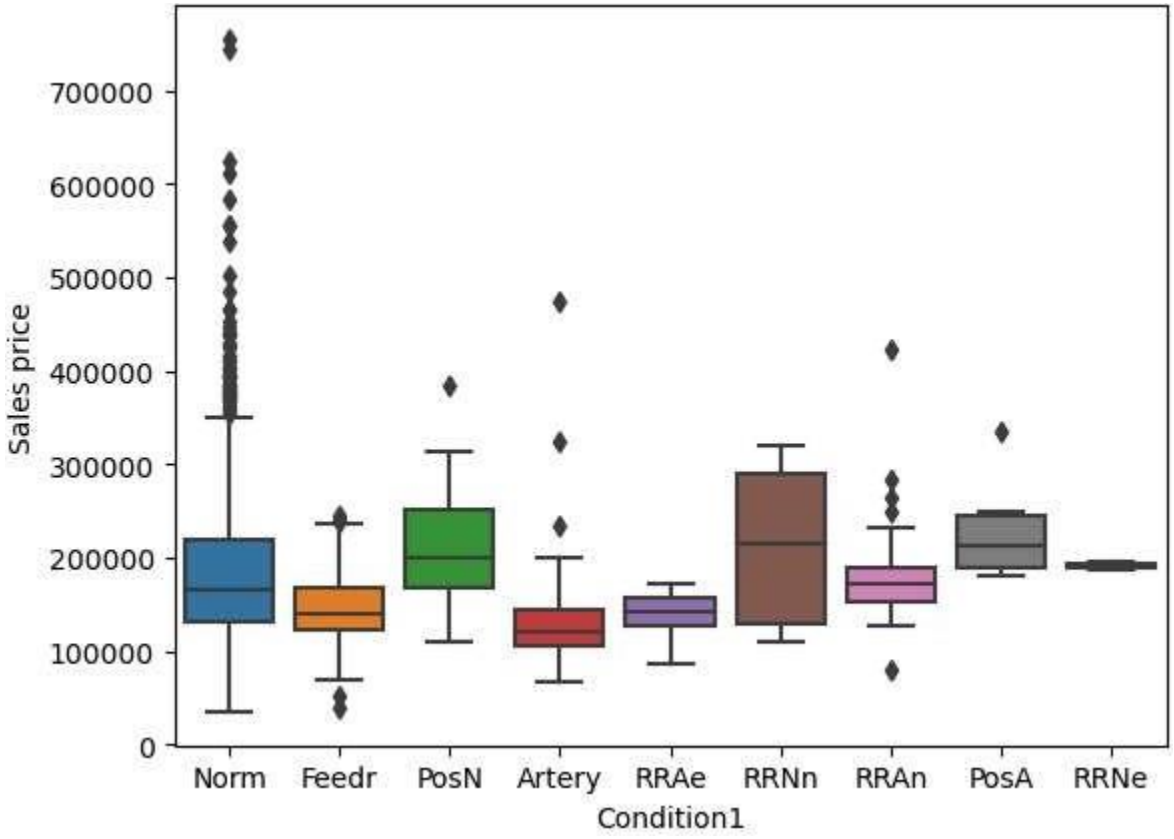

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Lvl
Bnk
Low
HLS
LandContour

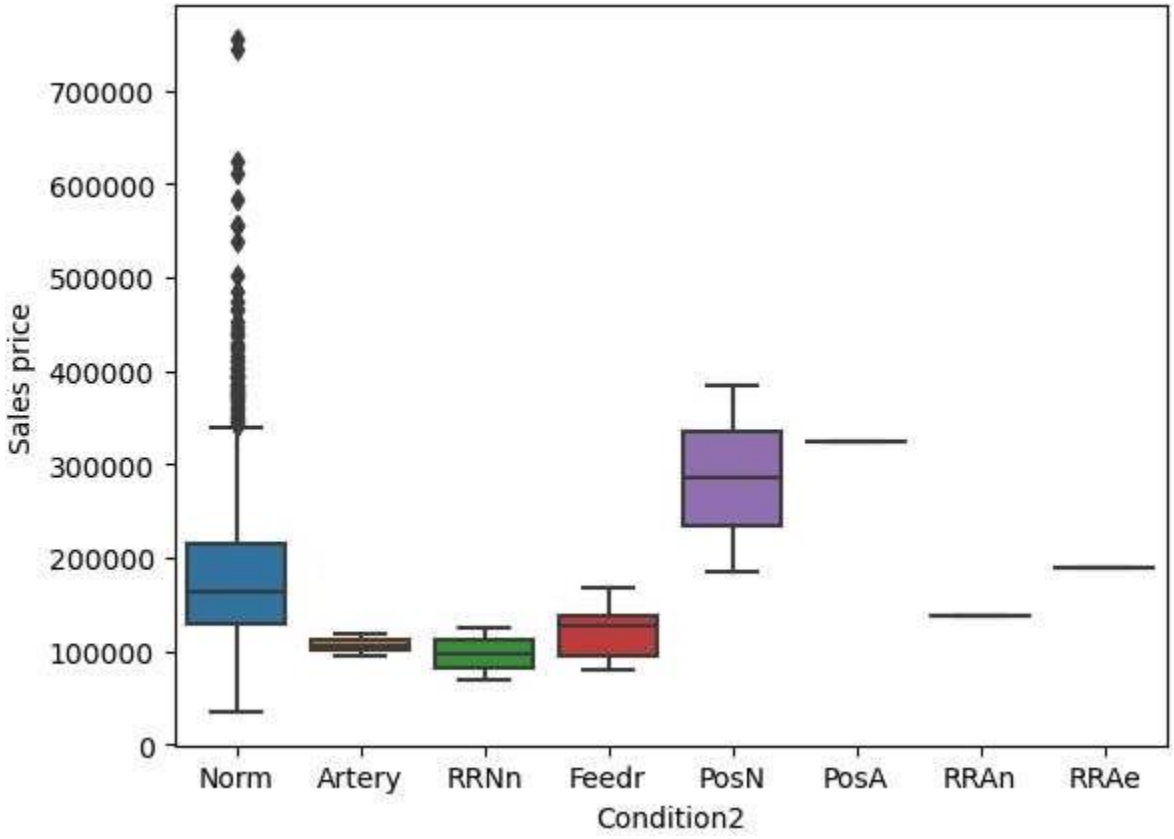

700000
600000
500000
400000
300000
200000
100000
0
Sales price
AllPub
NoSeWa
Utilities

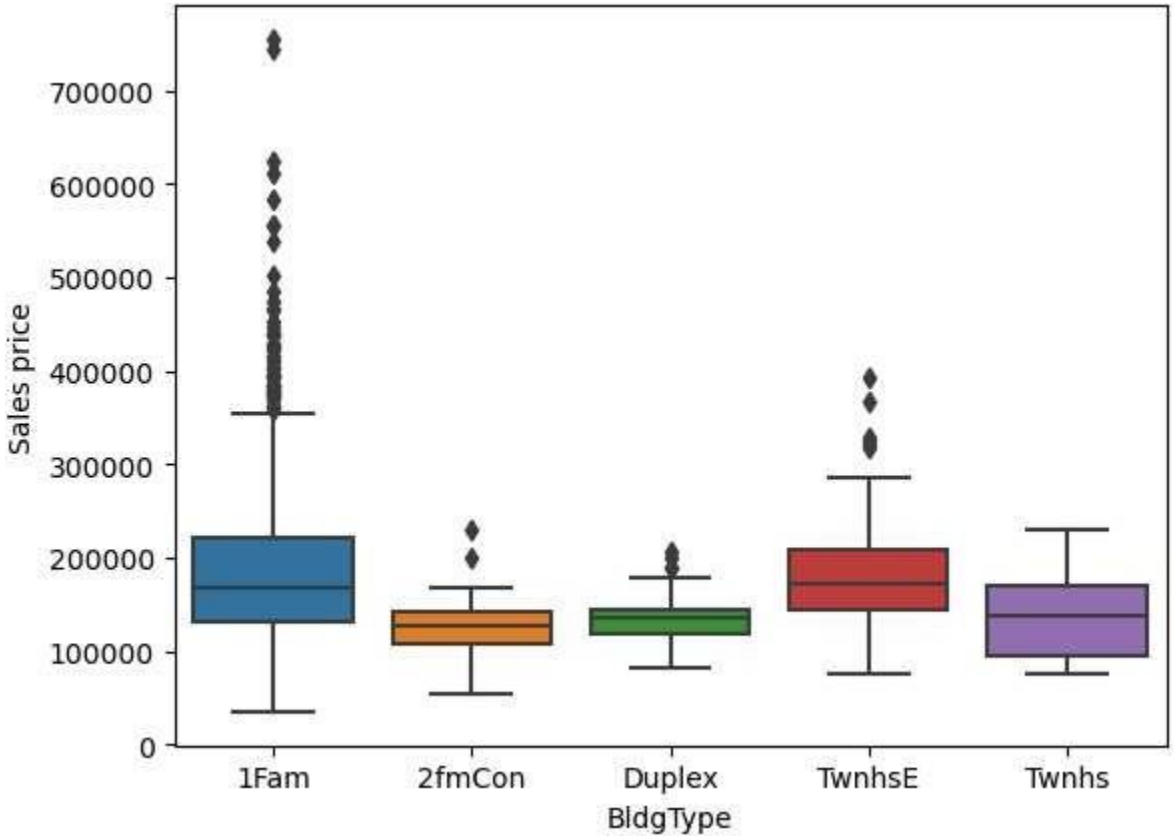

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Inside
FR2
Corner
CulDSac
FR3
LotConfig

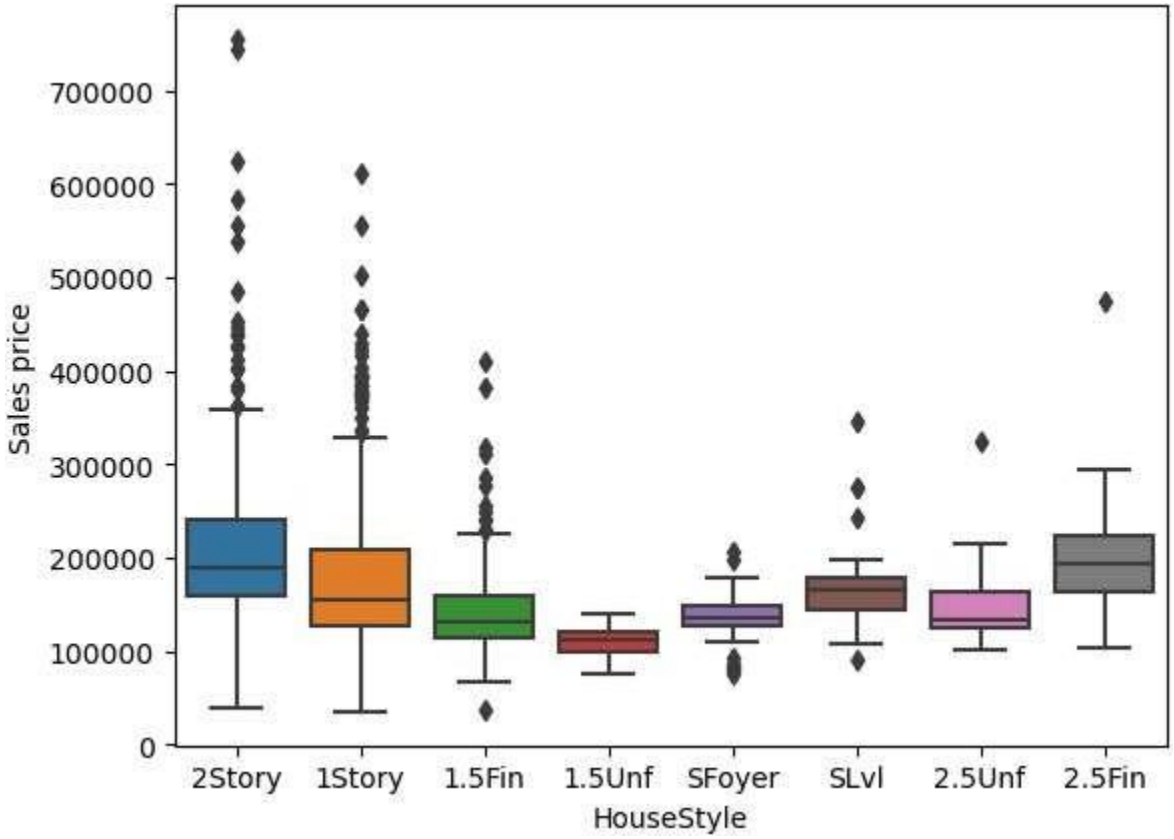

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Gtl
Mod
Sev
LandSlope

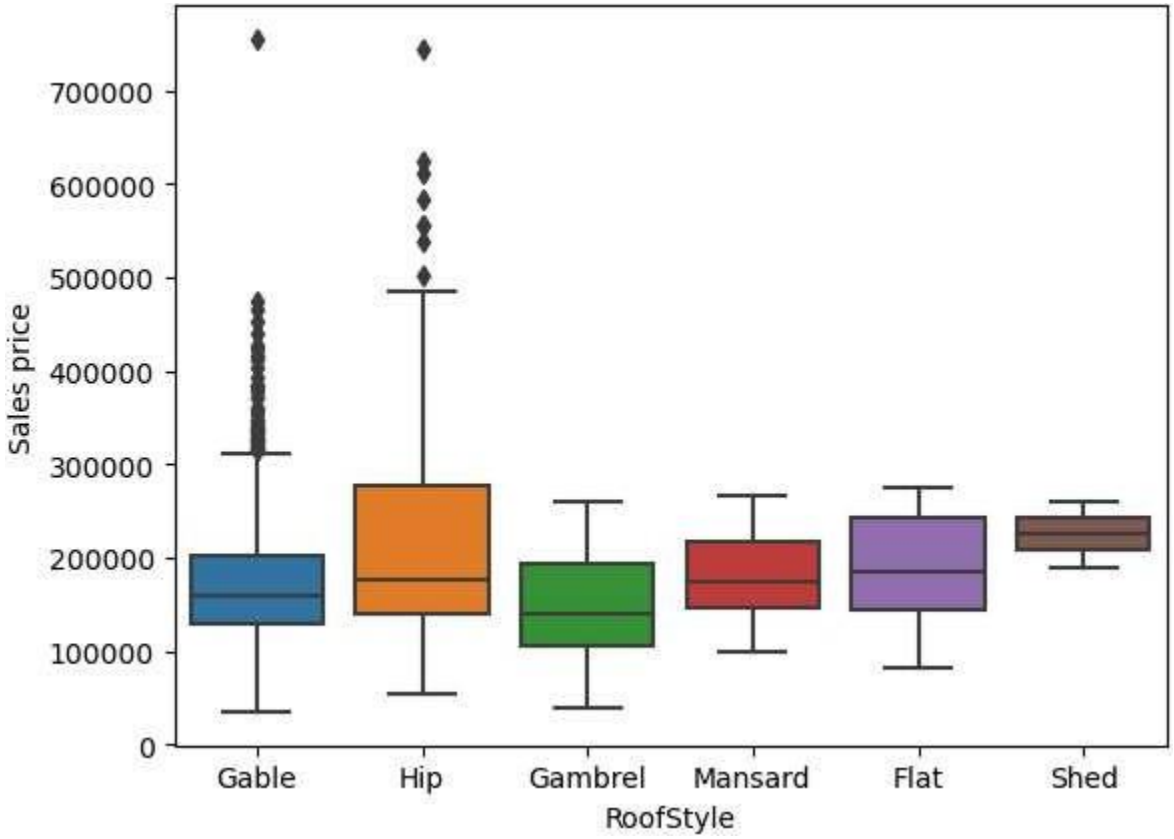

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Neighborhood

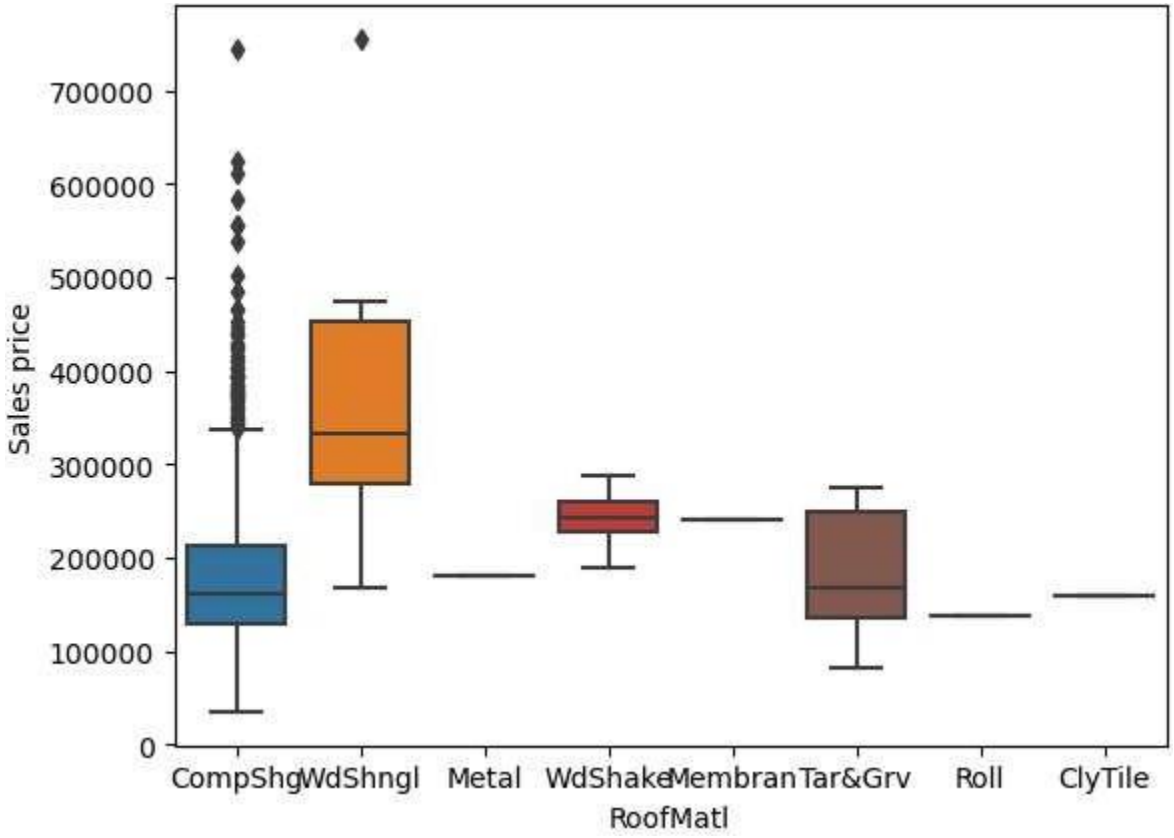

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Norm
Feedr
PosN
Artery
RRAe
RRNn
RRAn
PosA
RRNe
Condition1

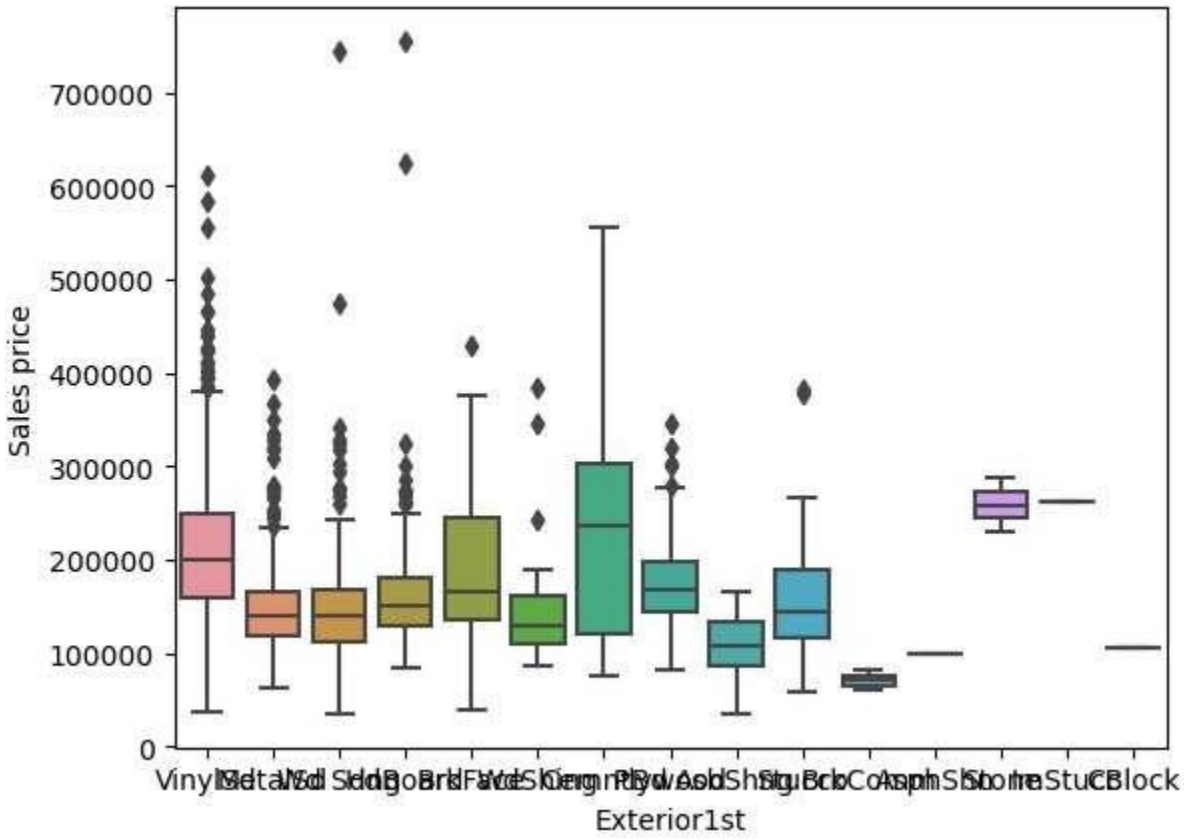

Sales price
700000
600000
500000
400000
300000
200000
100000
0
Norm
Artery
RRNn
Feedr
PosN
PosA
RRAn
RRAe
Condition2

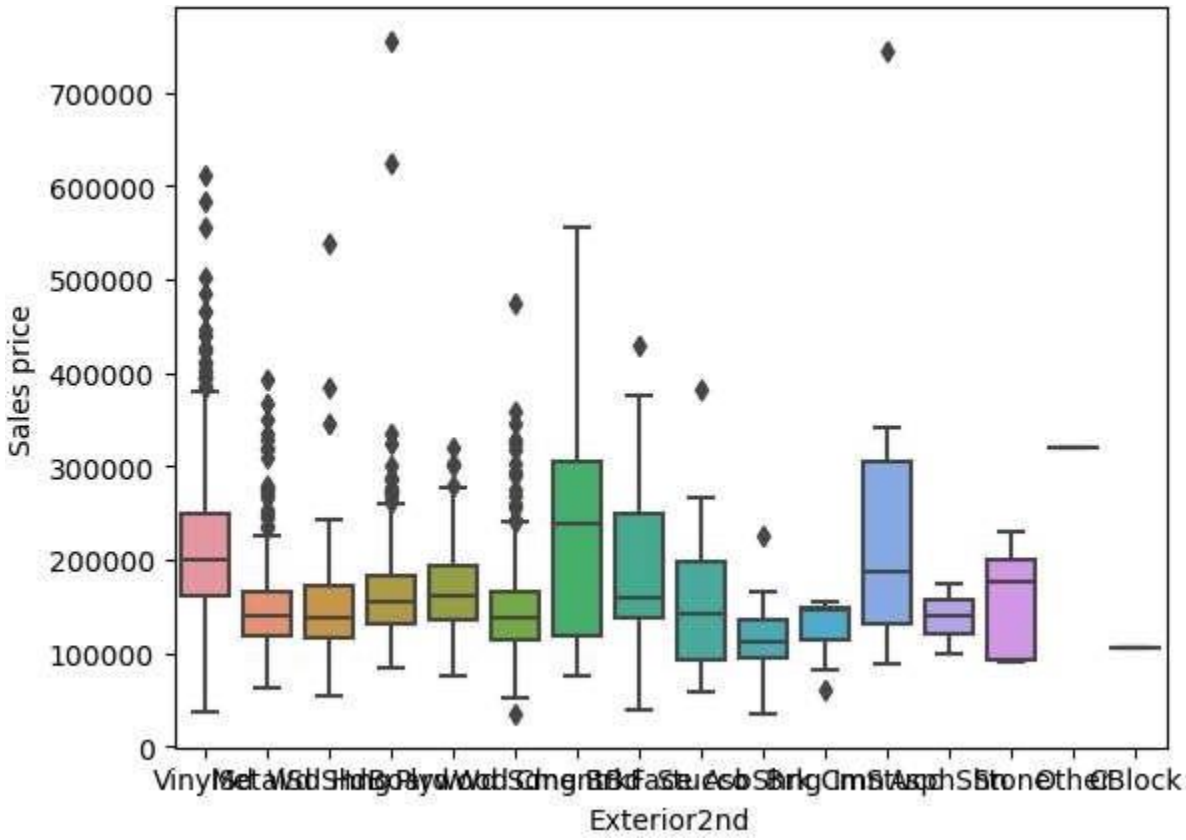

700000
600000
500000
400000
300000
200000
100000
0
Sales price
1Fam
2fmCon
Duplex
TwnhsE
Twnhs
BldgType

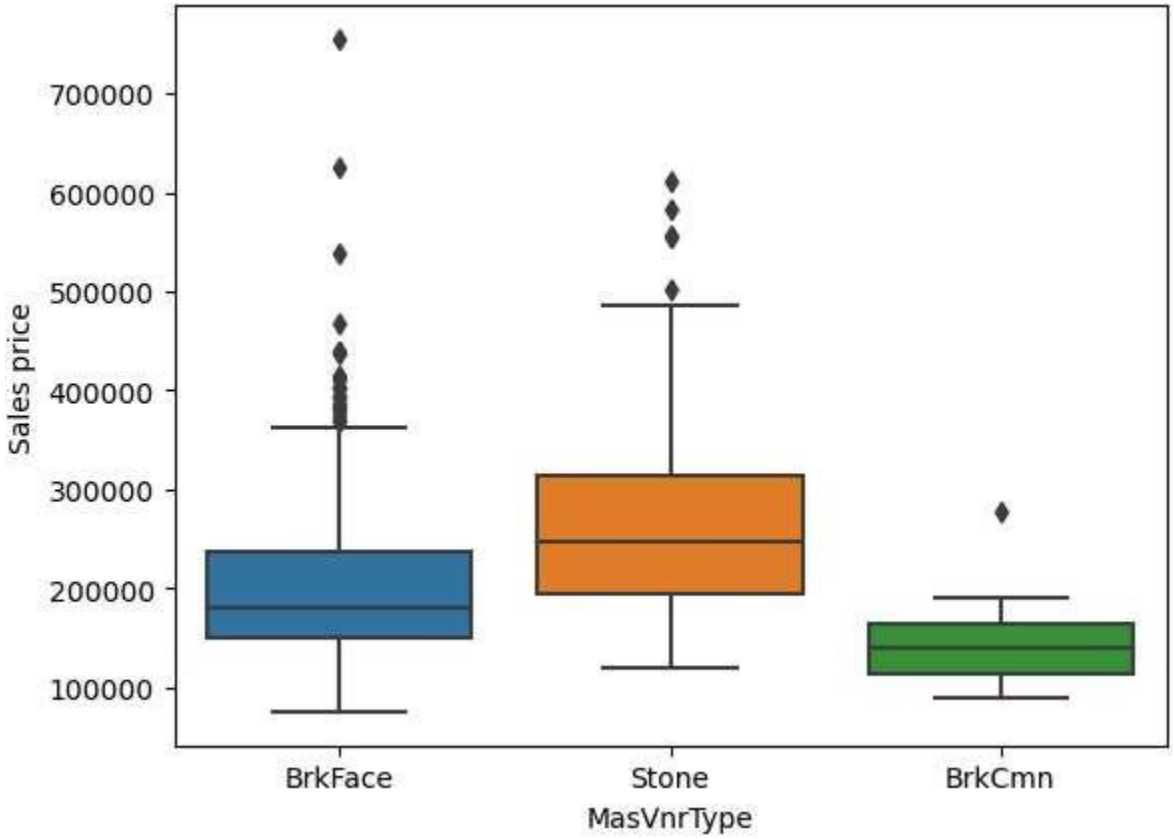

Sales price
700000
600000
500000
400000
300000
200000
100000
0
2Story
1Story
1.5Fin
1.5Unf
SFoyer
SLvl
2.5Unf
2.5Fin
HouseStyle

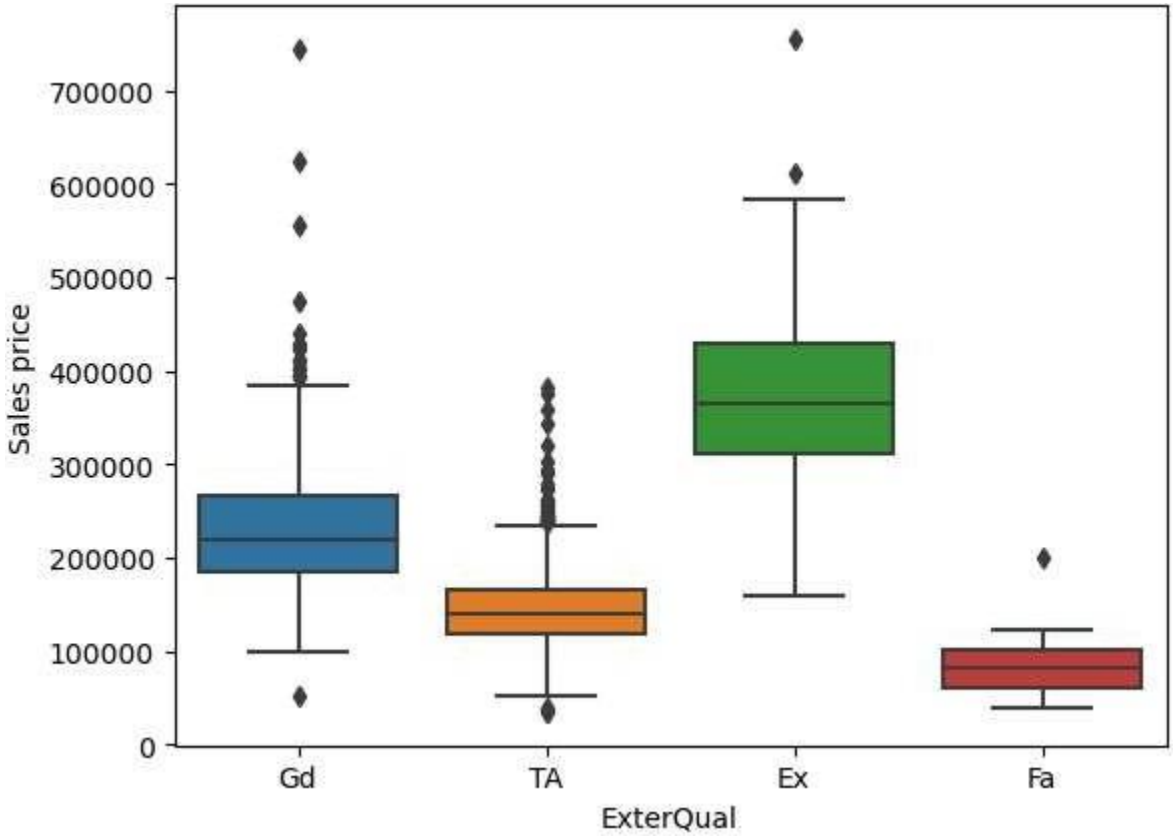

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Gable
Hip
Gambrel
Mansard
Flat
Shed
RoofStyle

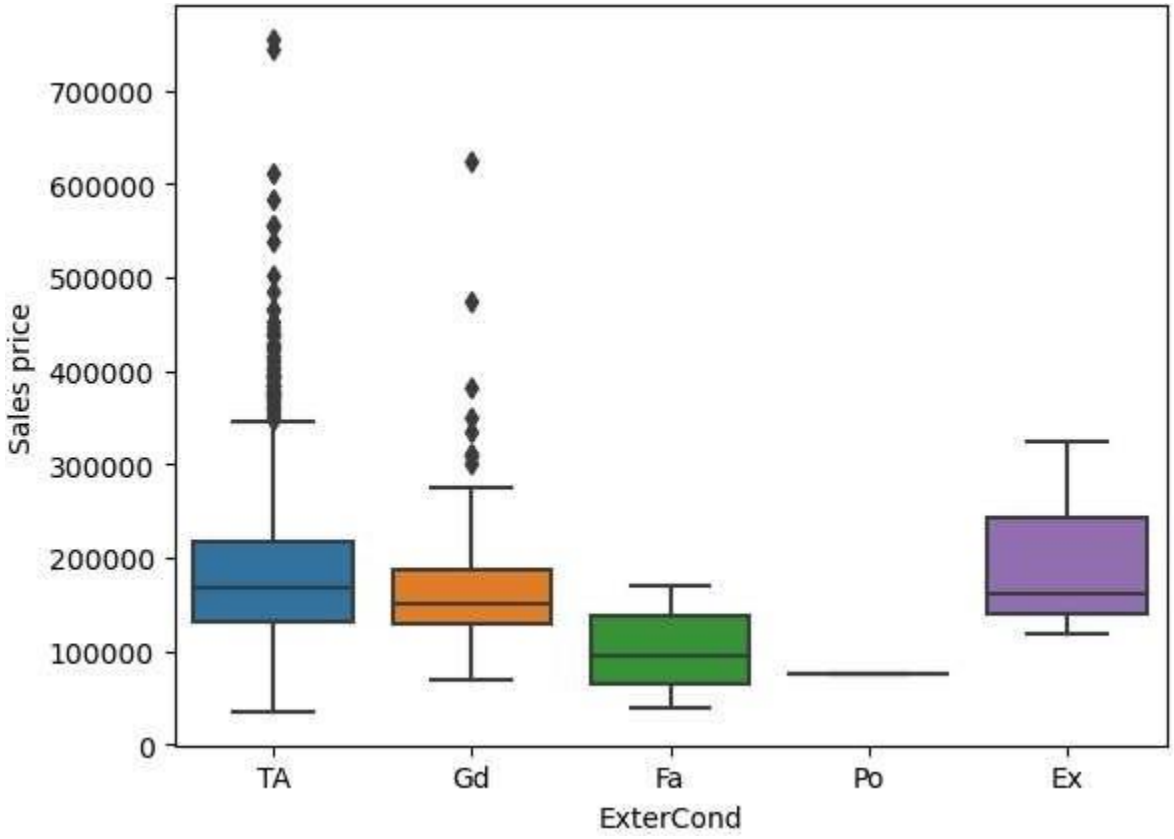

Sales price
700000
600000
500000
400000
300000
200000
100000
0
CompShg
WdShngl
Metal
WdShake
Membran
Tar&Grv
Roll
ClyTile
RoofMatl

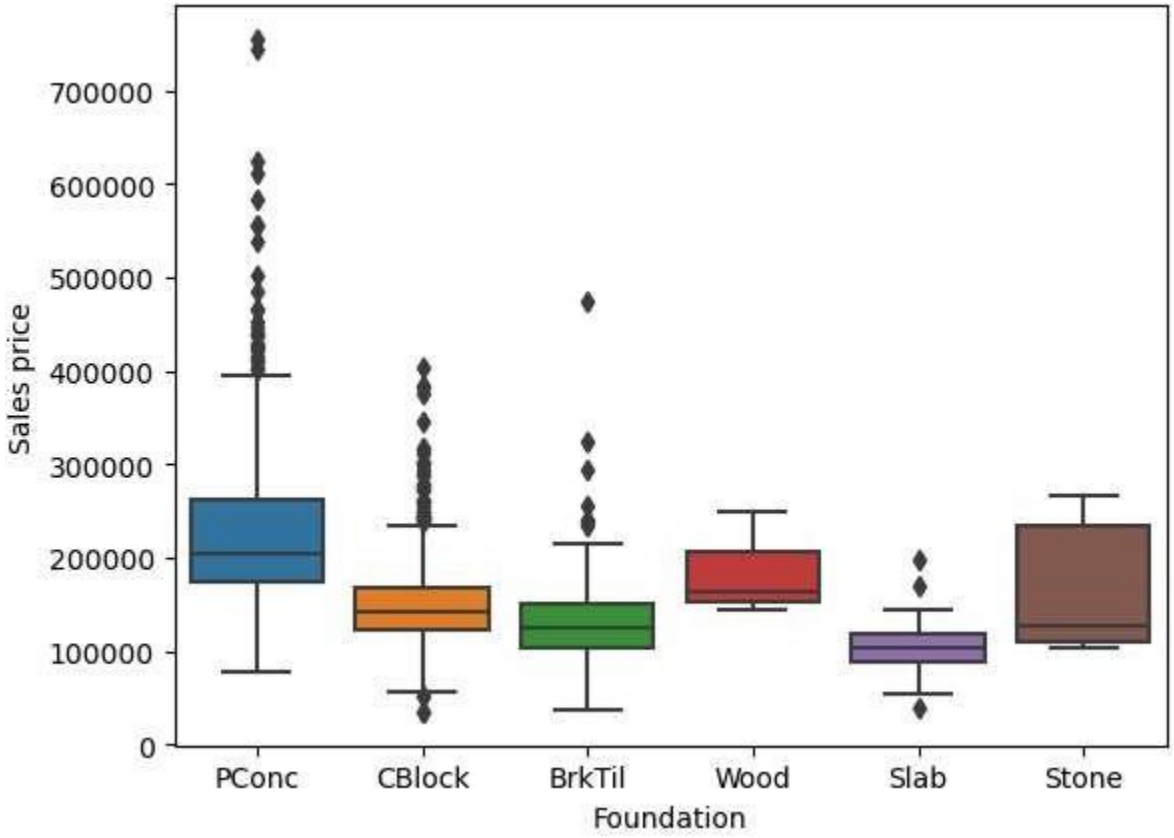

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Exterior1st

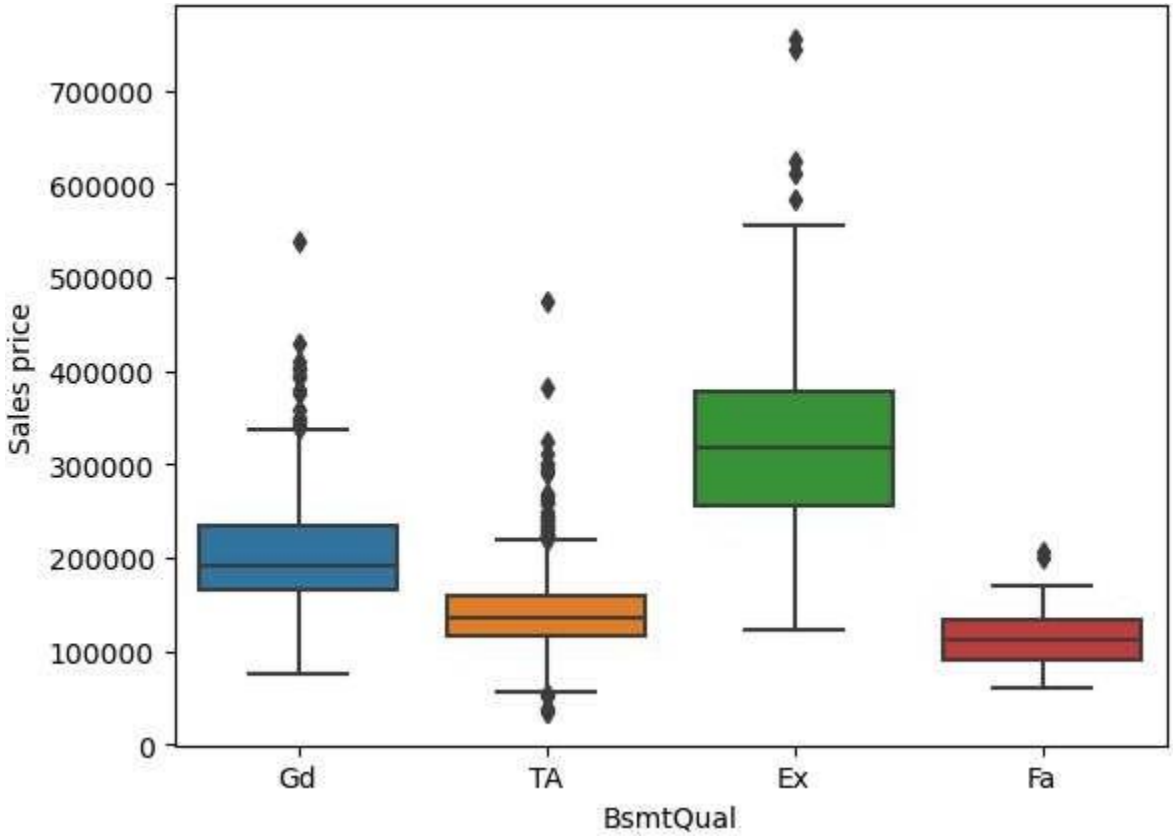

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Exterior2nd

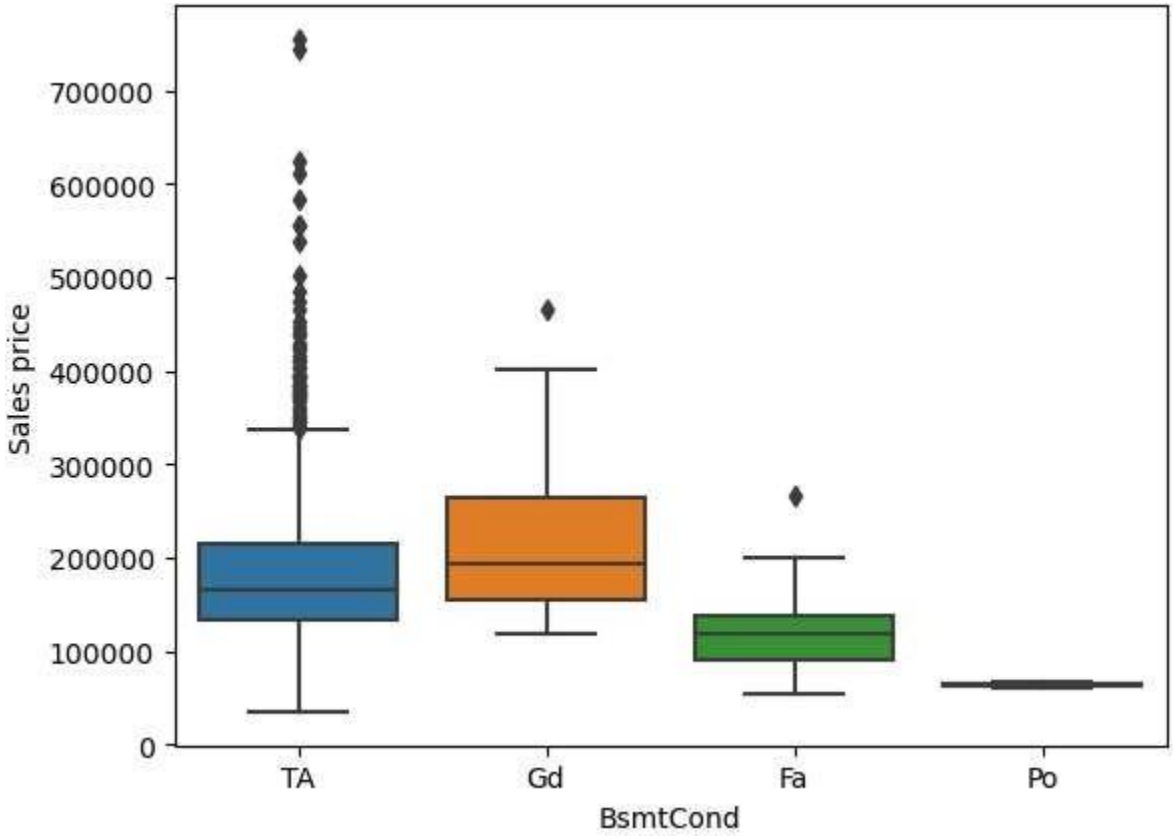

700000
600000
500000
400000
300000
200000
100000
Sales price
BrkFace
Stone
BrkCmn
MasVnrType

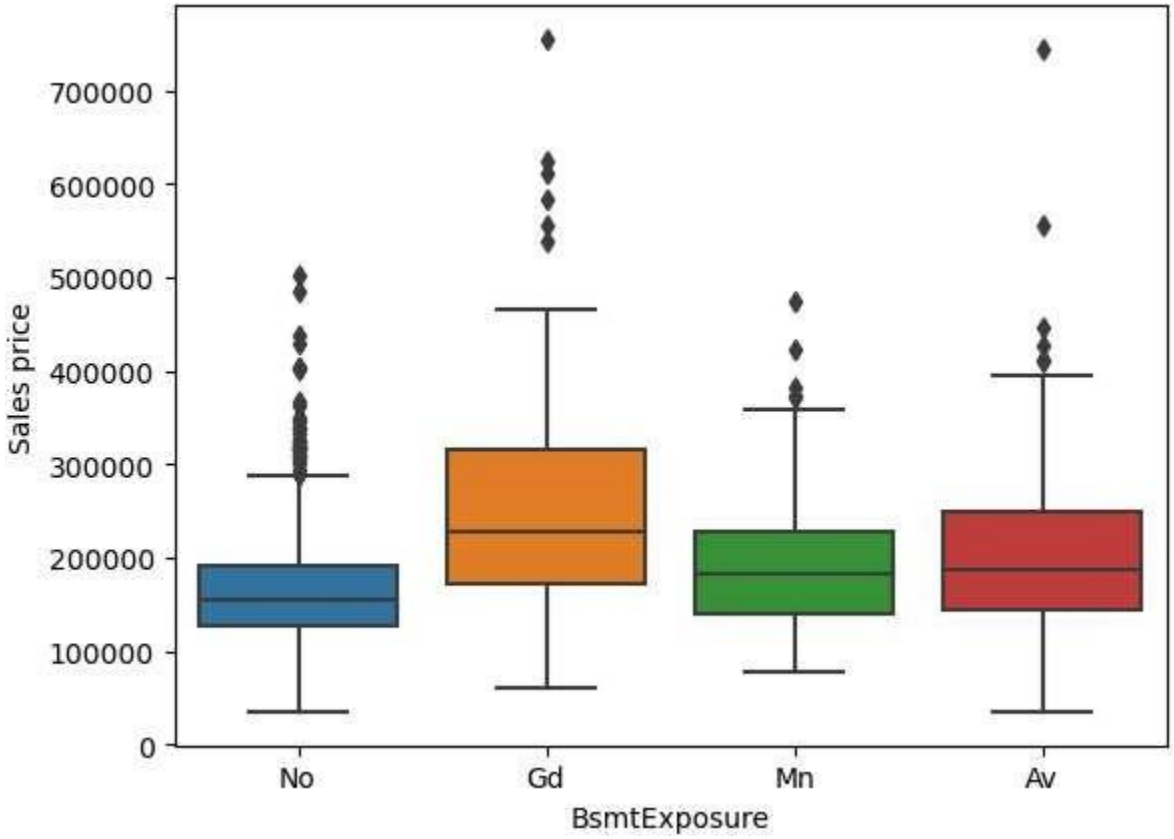

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Gd
TA
Ex
Fa
ExterQual

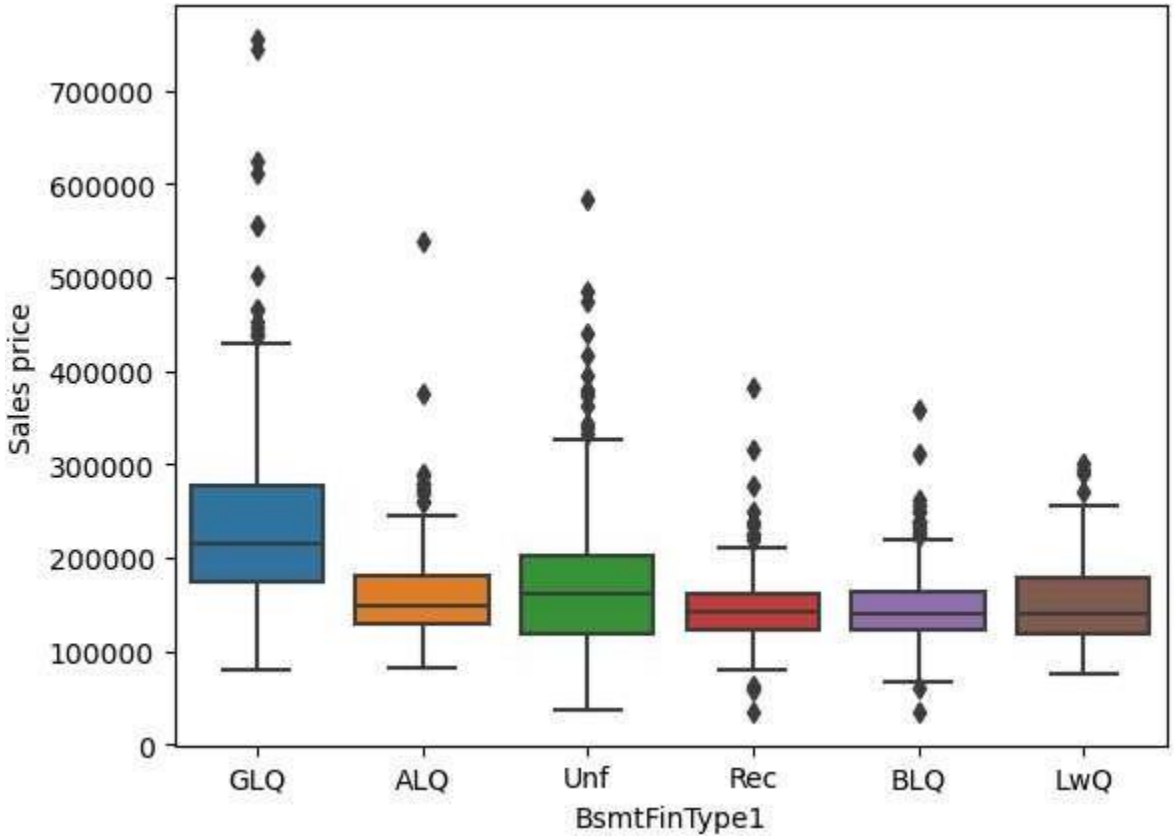

700000
600000
500000
400000
300000
200000
100000
0
Sales price
TA
Gd
Fa
Po
Ex
ExterCond

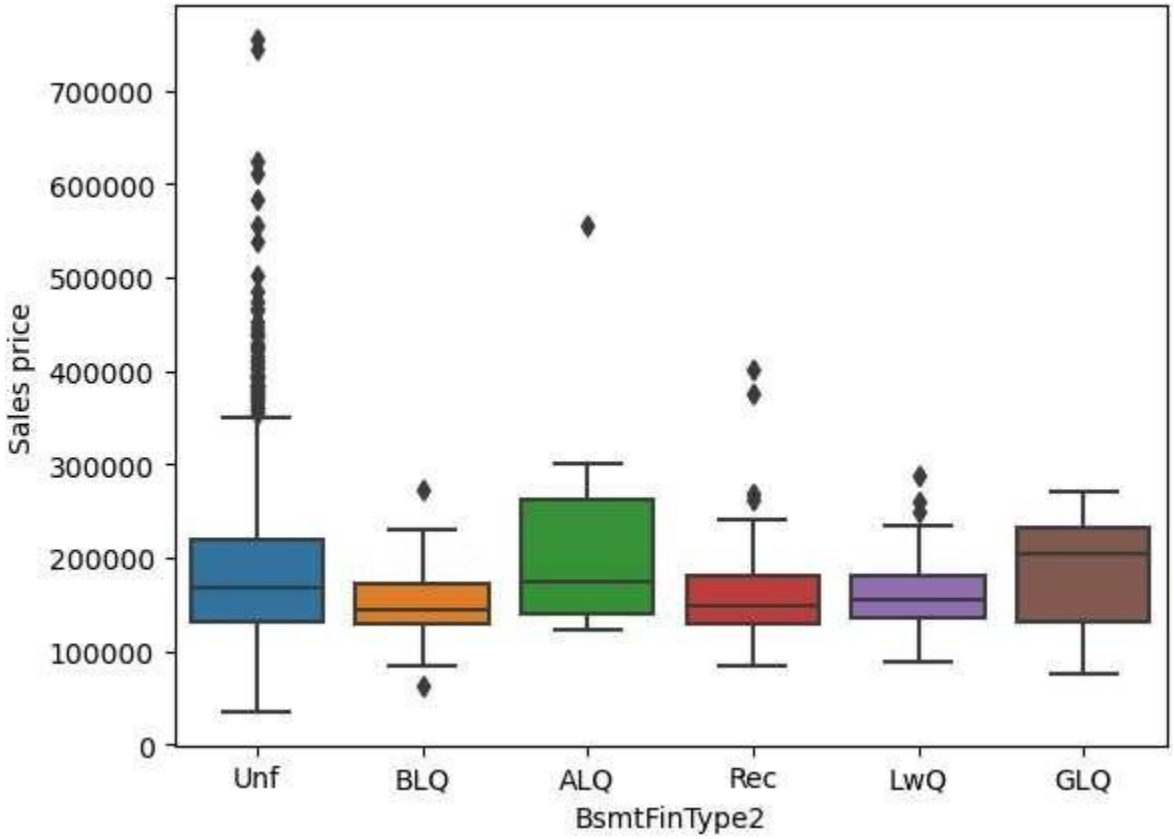

700000
600000
500000
400000
300000
200000
100000
0
Sales price
PConc
CBlock
BrkTil
Wood
Slab
Stone
Foundation

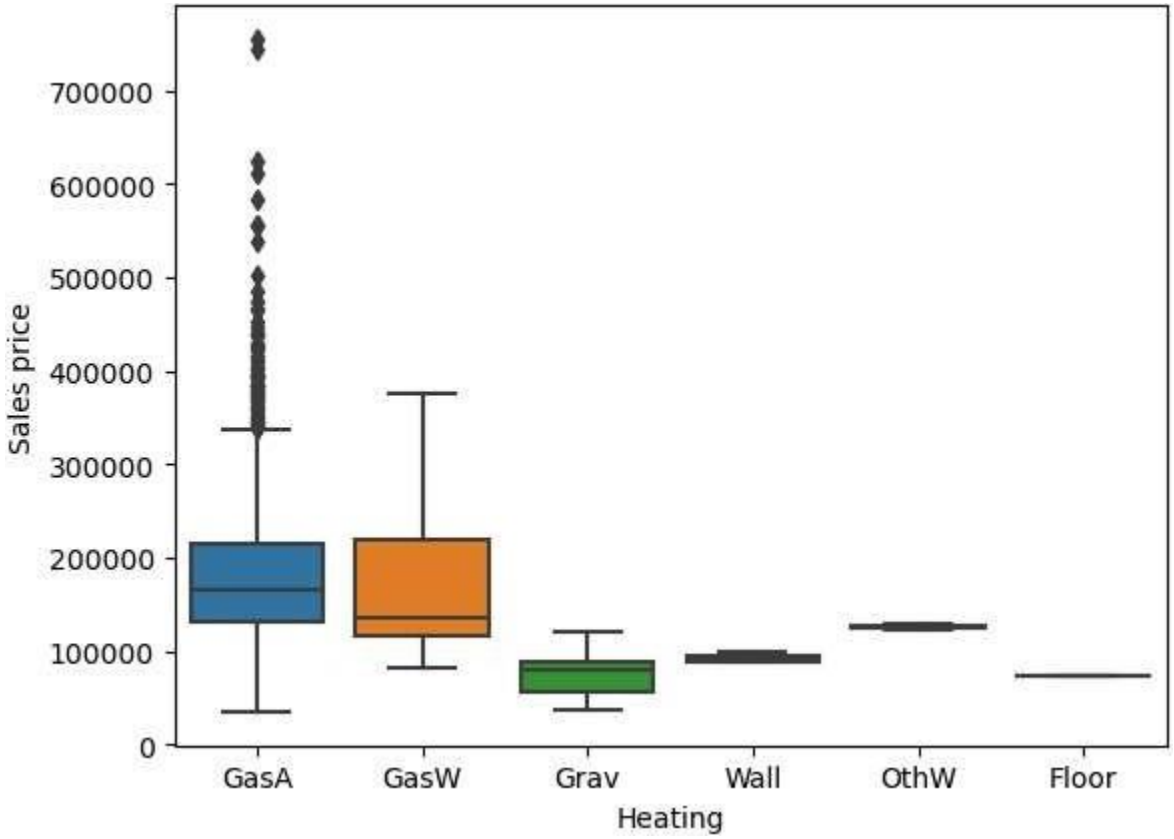

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Gd
TA
Ex
Fa
BsmtQual

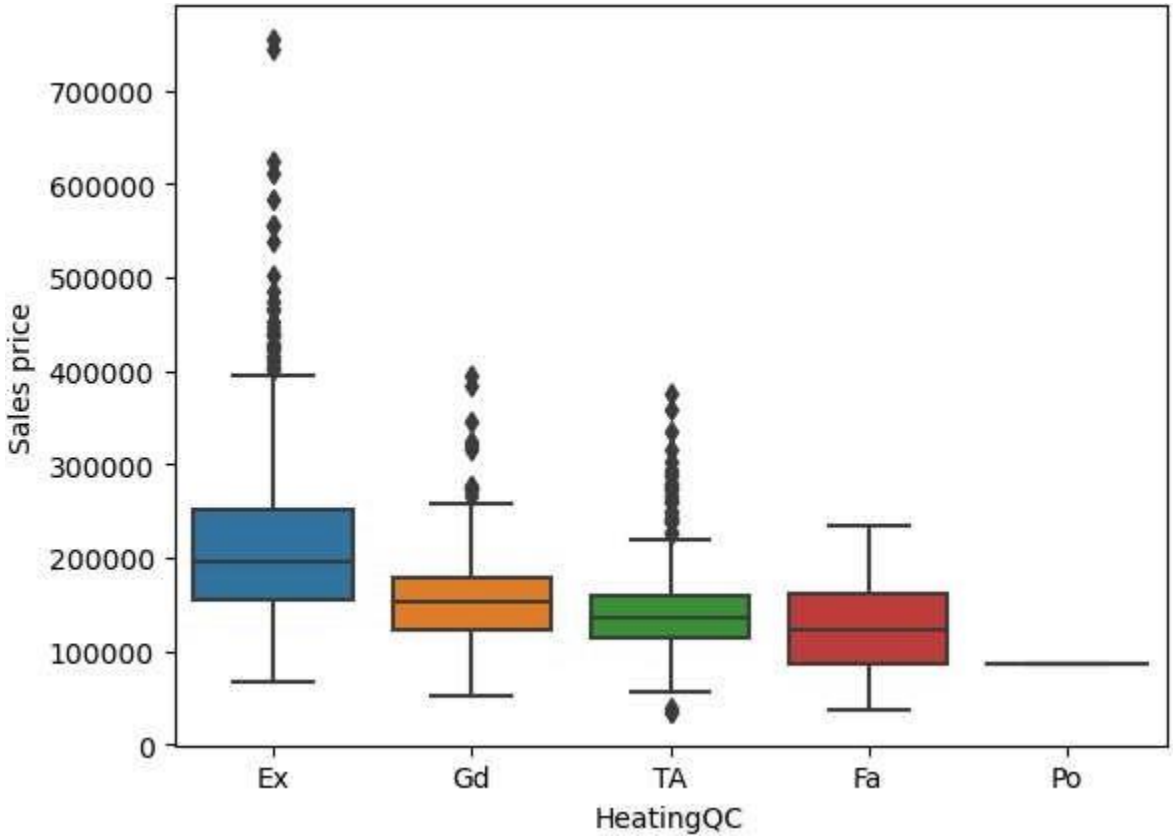

700000
600000
500000
400000
300000
200000
100000
0
Sales price
TA
Gd
Fa
Po
BsmtCond

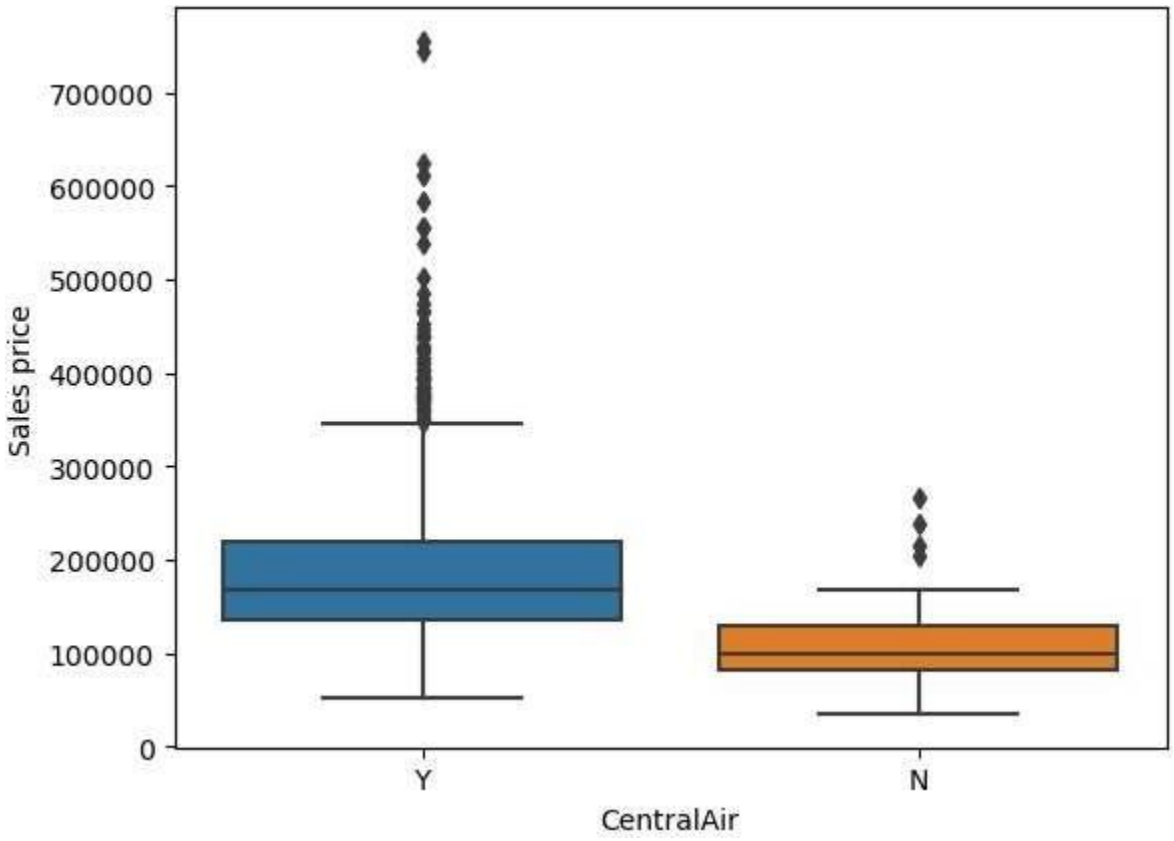

700000
600000
500000
400000
300000
200000
100000
0
Sales price
No
Gd
Mn
Av
BsmtExposure

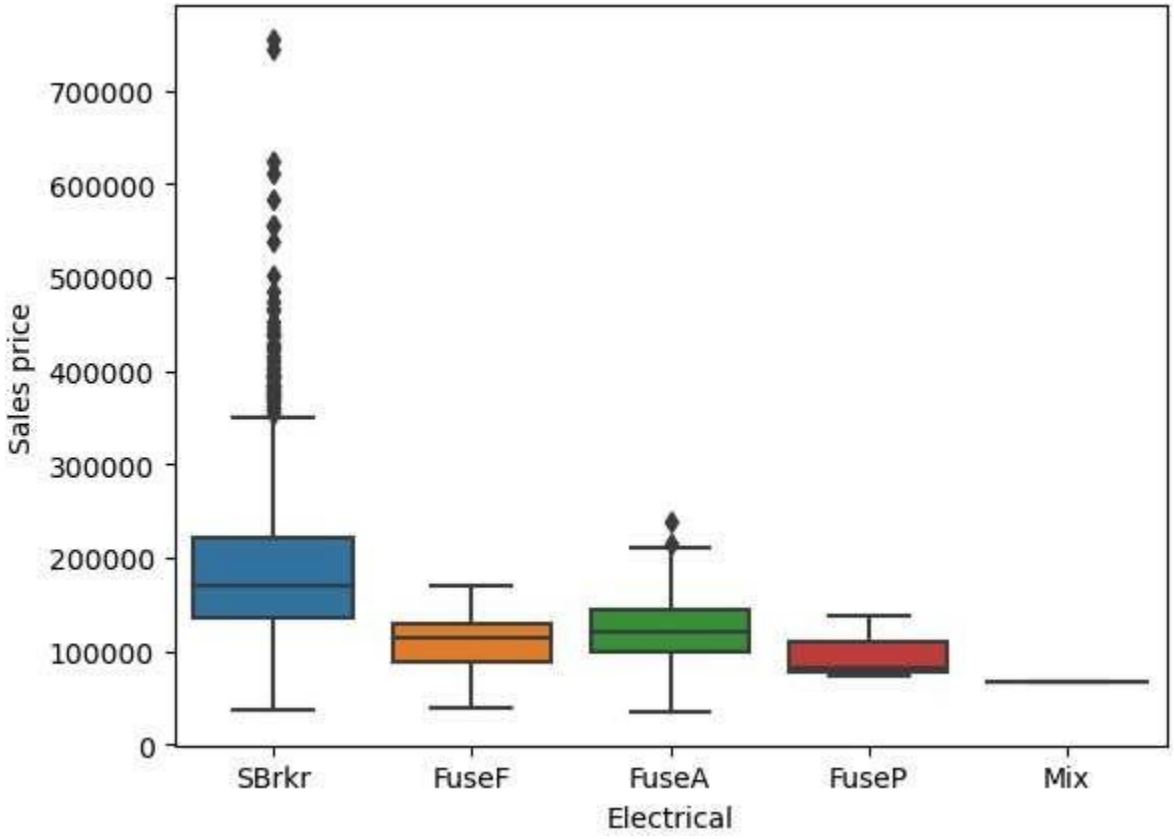

700000
600000
500000
400000
300000
200000
100000
0
Sales price
GLQ
ALQ
Unf
Rec
BLQ
LwQ
BsmtFinType1

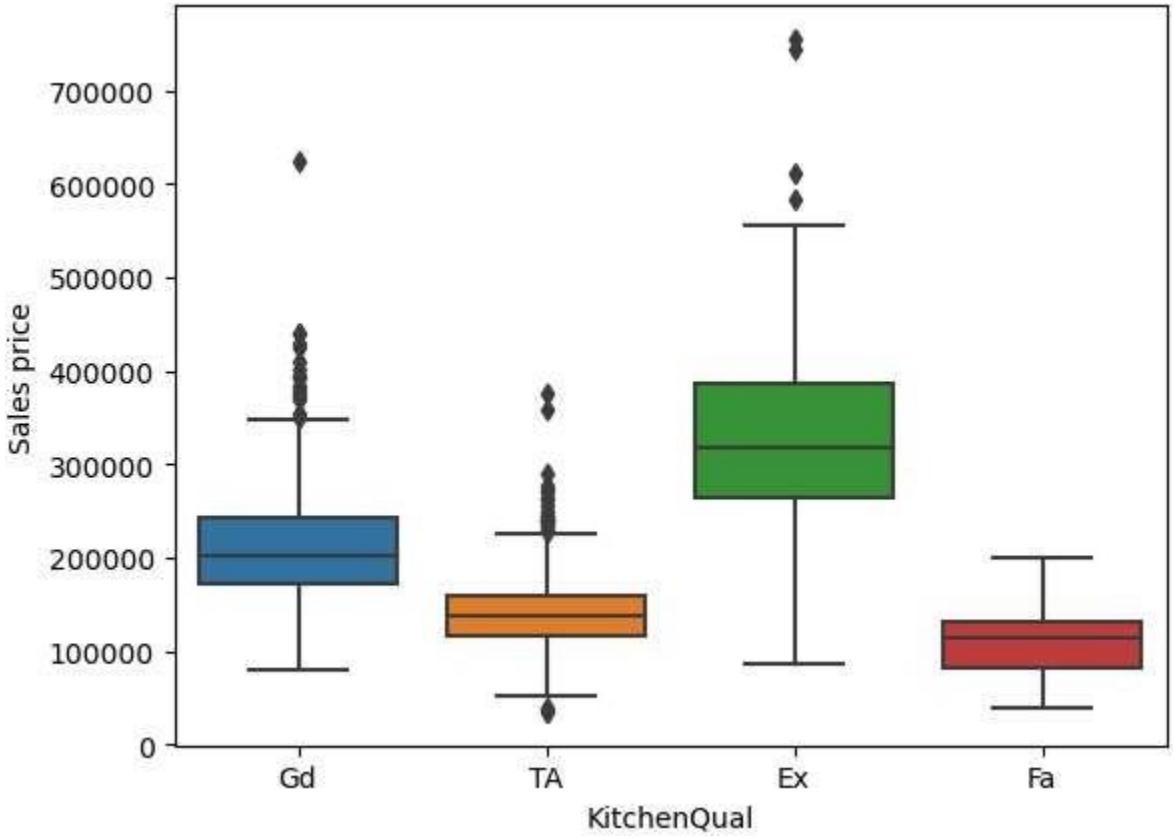

Sales price
700000
600000
500000
400000
300000
200000
100000
0
Unf
BLQ
ALQ
Rec
LwQ
GLQ
BsmtFinType2

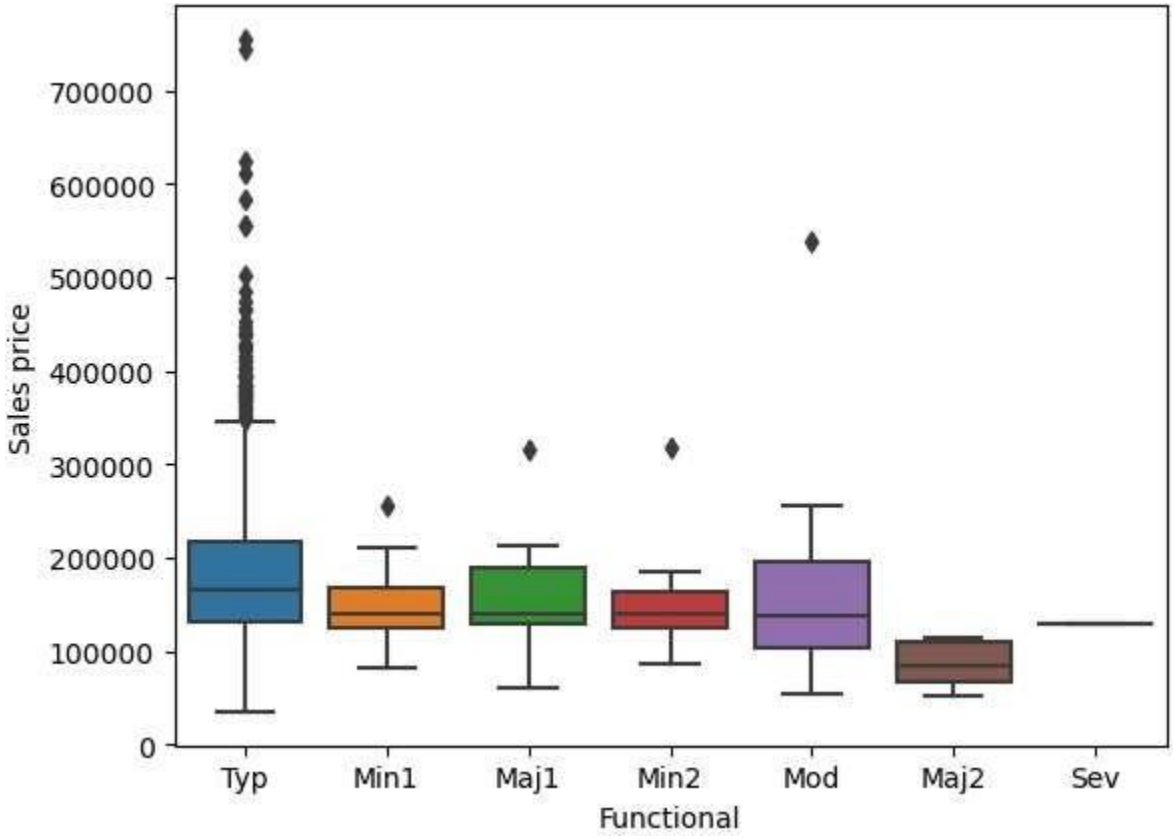

700000
600000
500000
400000
300000
200000
100000
0
Sales price
GasA
GasW
Grav
Wall
OthW
Floor
Heating

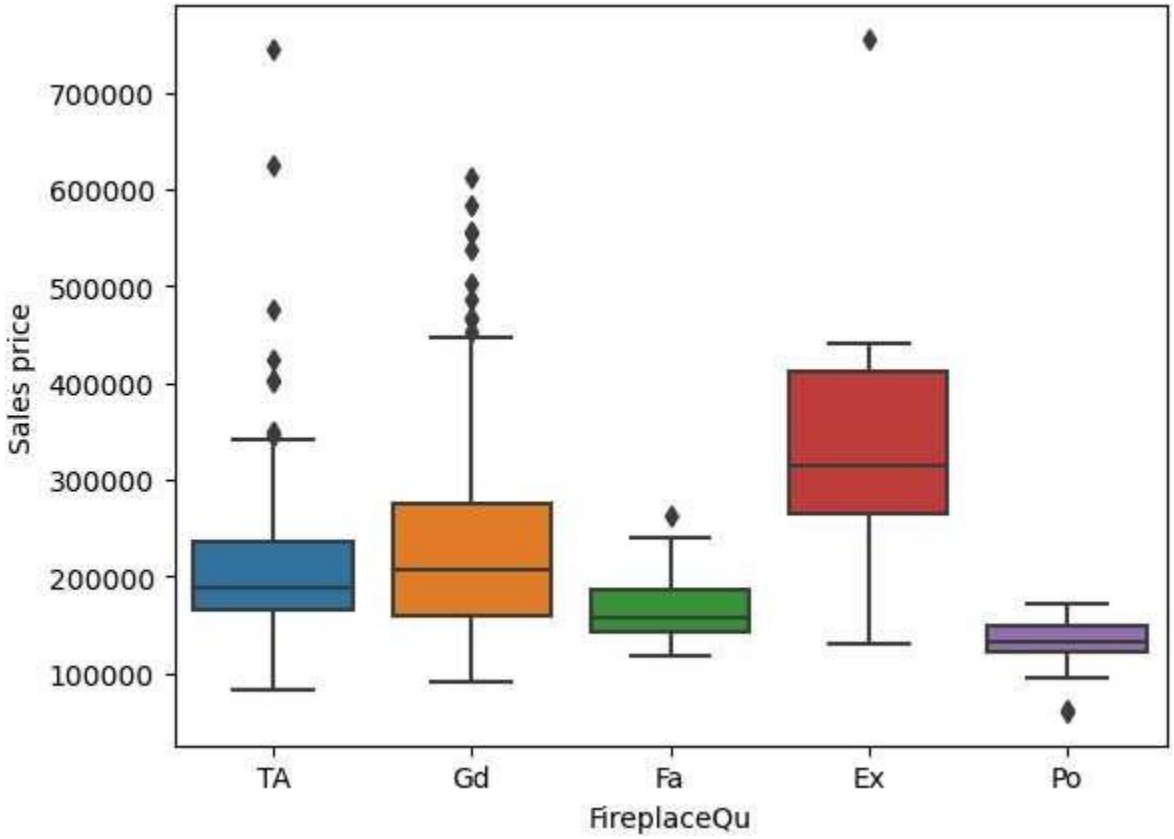

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Ex
Gd
TA
Fa
Po
HeatingQC

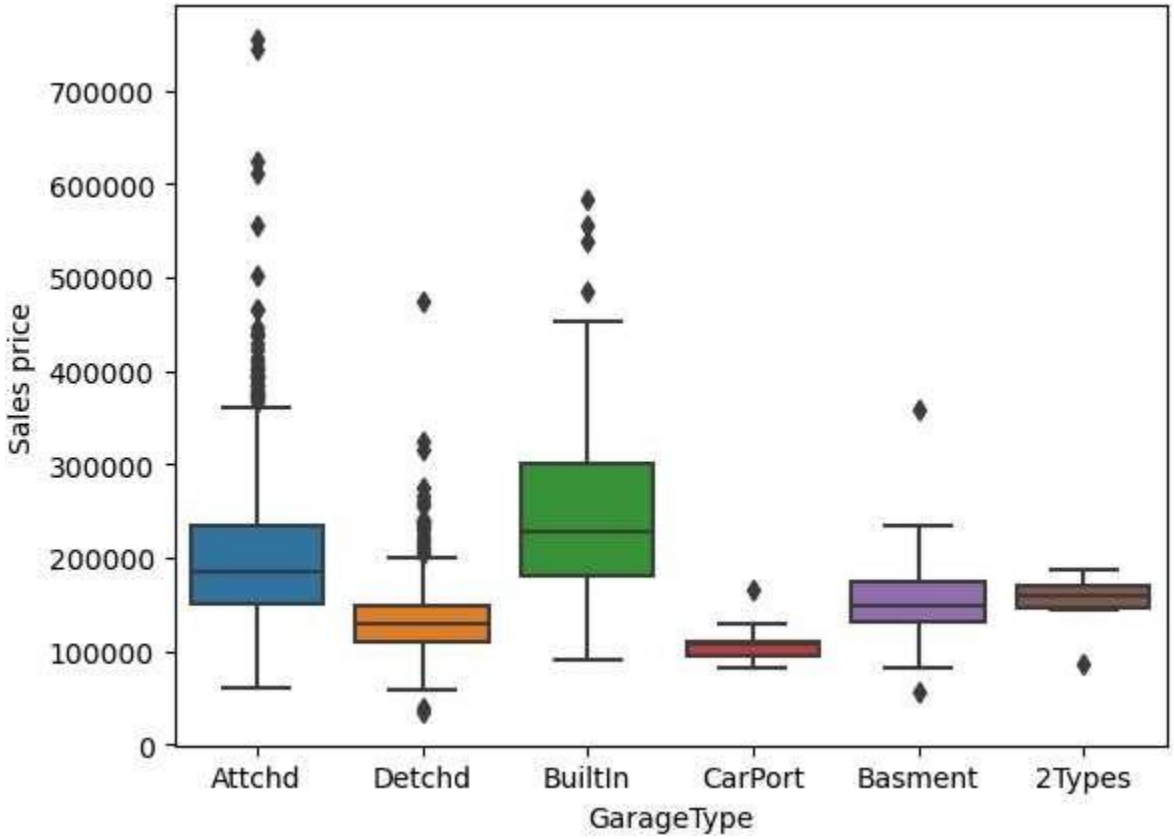

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Y
N
CentralAir

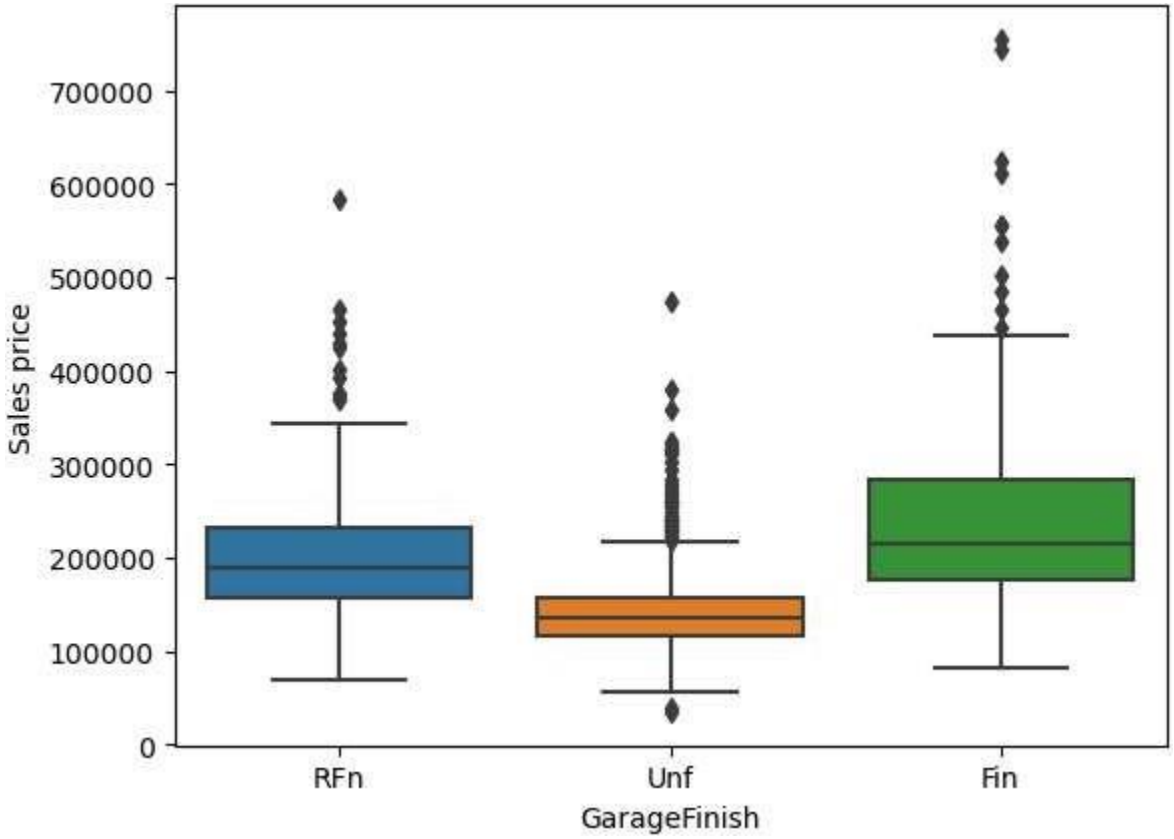

700000
600000
500000
400000
300000
200000
100000
0
Sales price
SBrkr
FuseF
FuseA
FuseP
Mix
Electrical

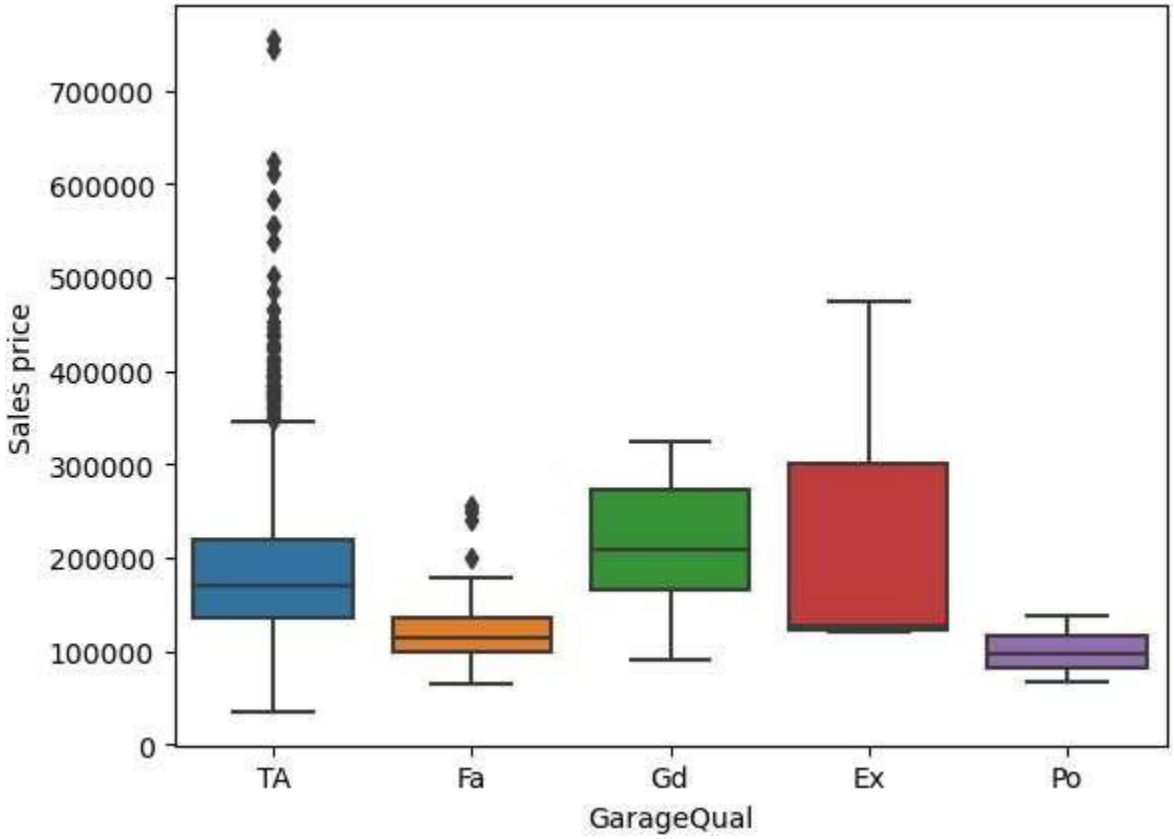

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Gd
TA
Ex
Fa
KitchenQual

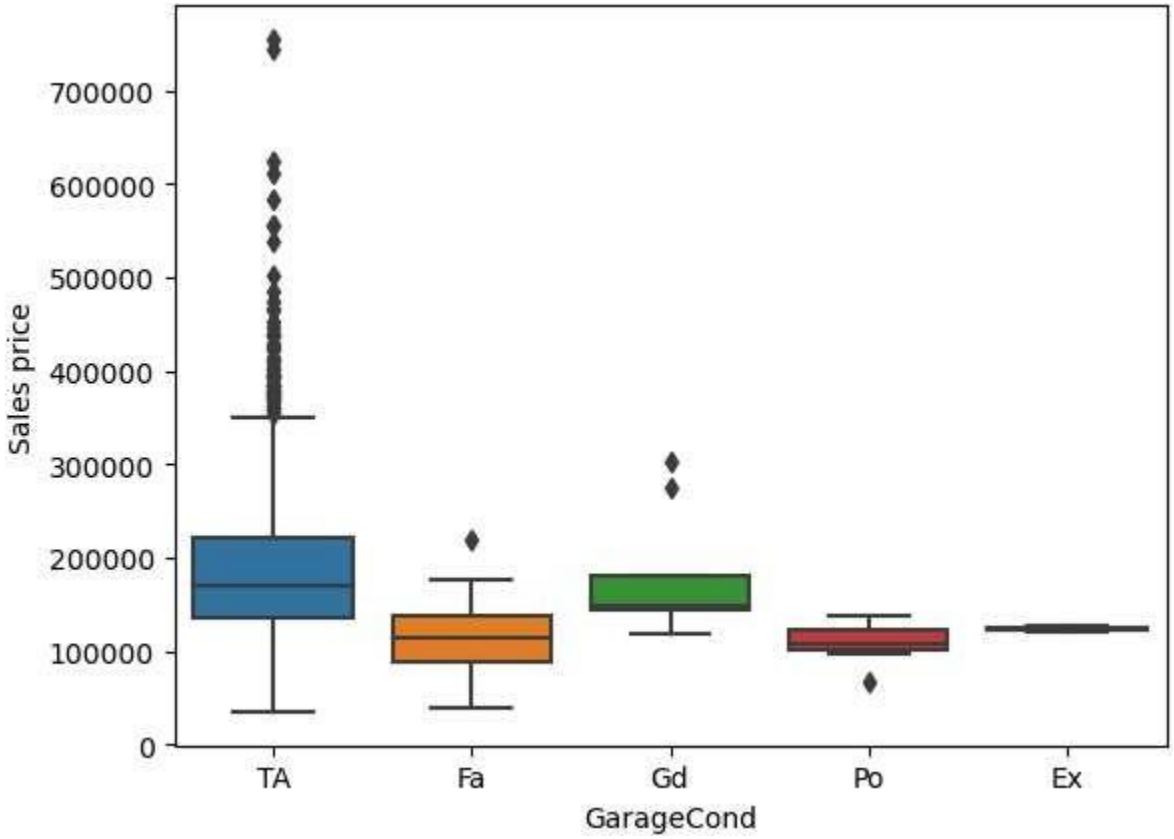

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Typ
Min1
Maj1
Min2
Mod
Maj2
Sev
Functional

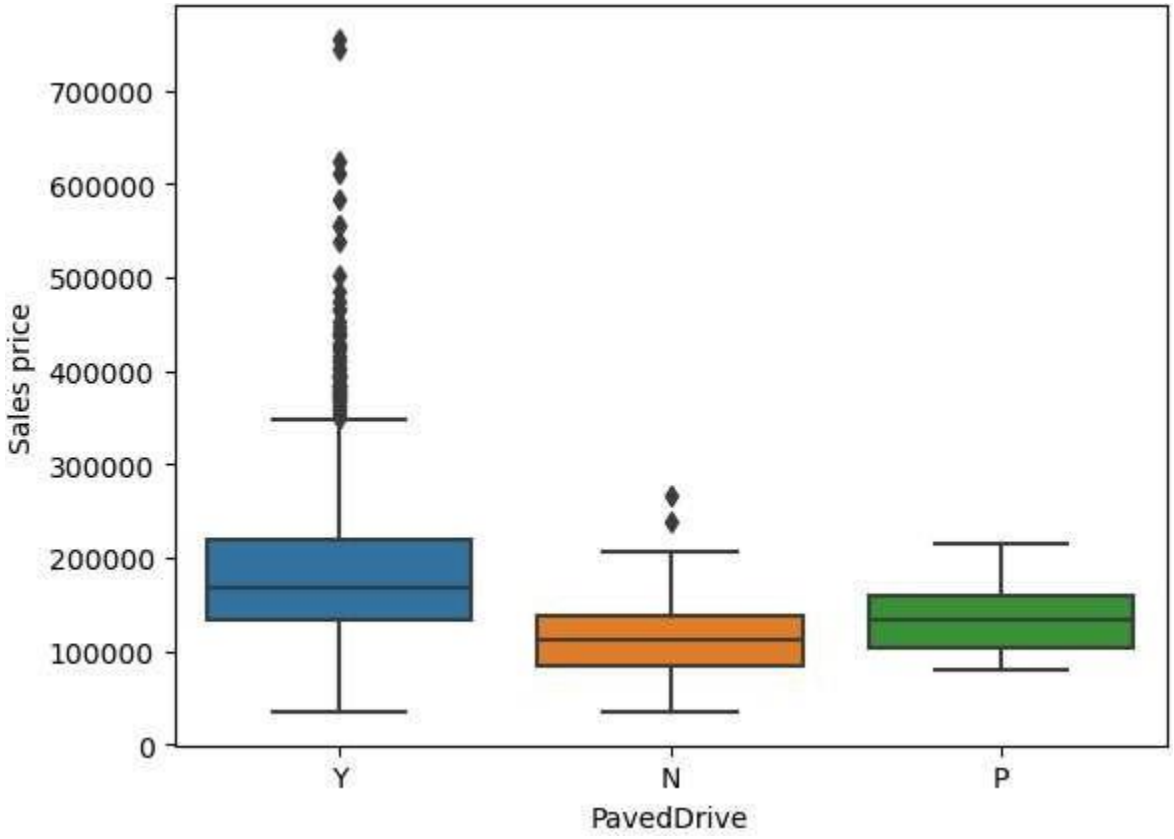

700000
600000
500000
400000
300000
200000
100000
Sales price
TA
Gd
Fa
Ex
Po
FireplaceQu

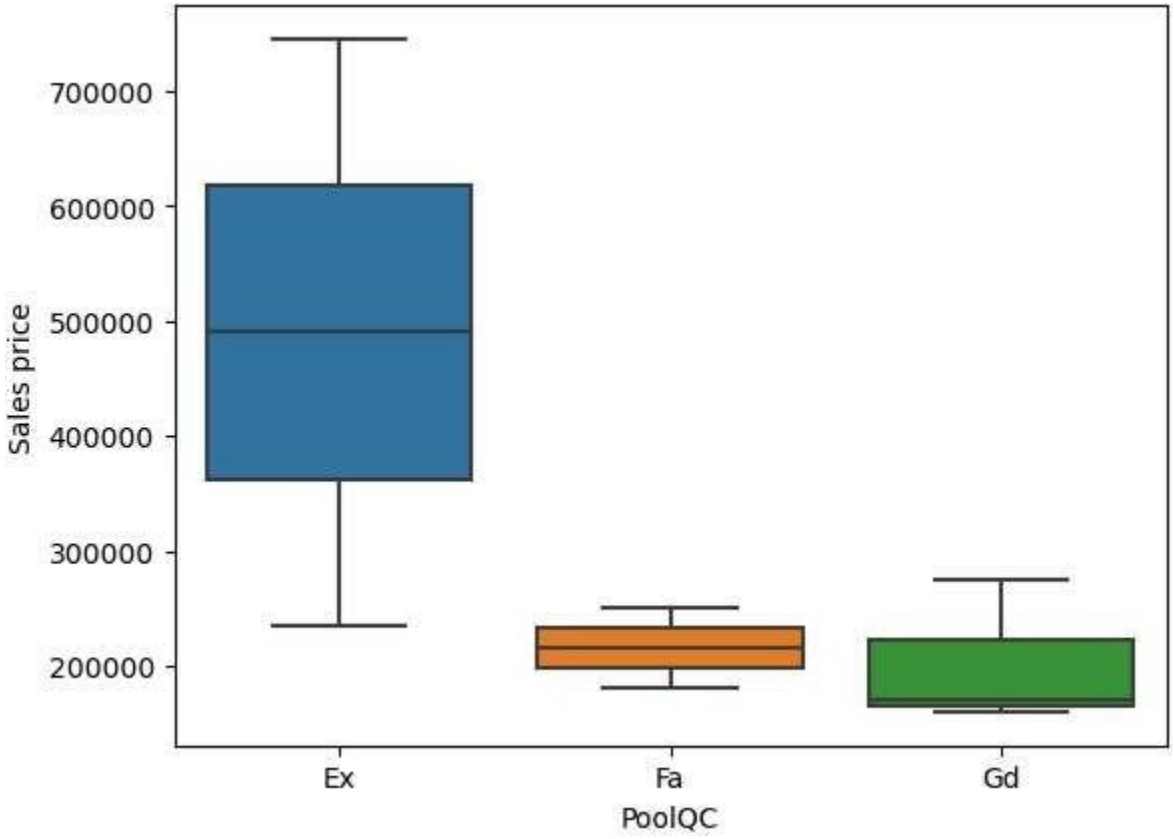

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Attchd
Detchd
BuiltIn
CarPort
Basment
2Types
GarageType

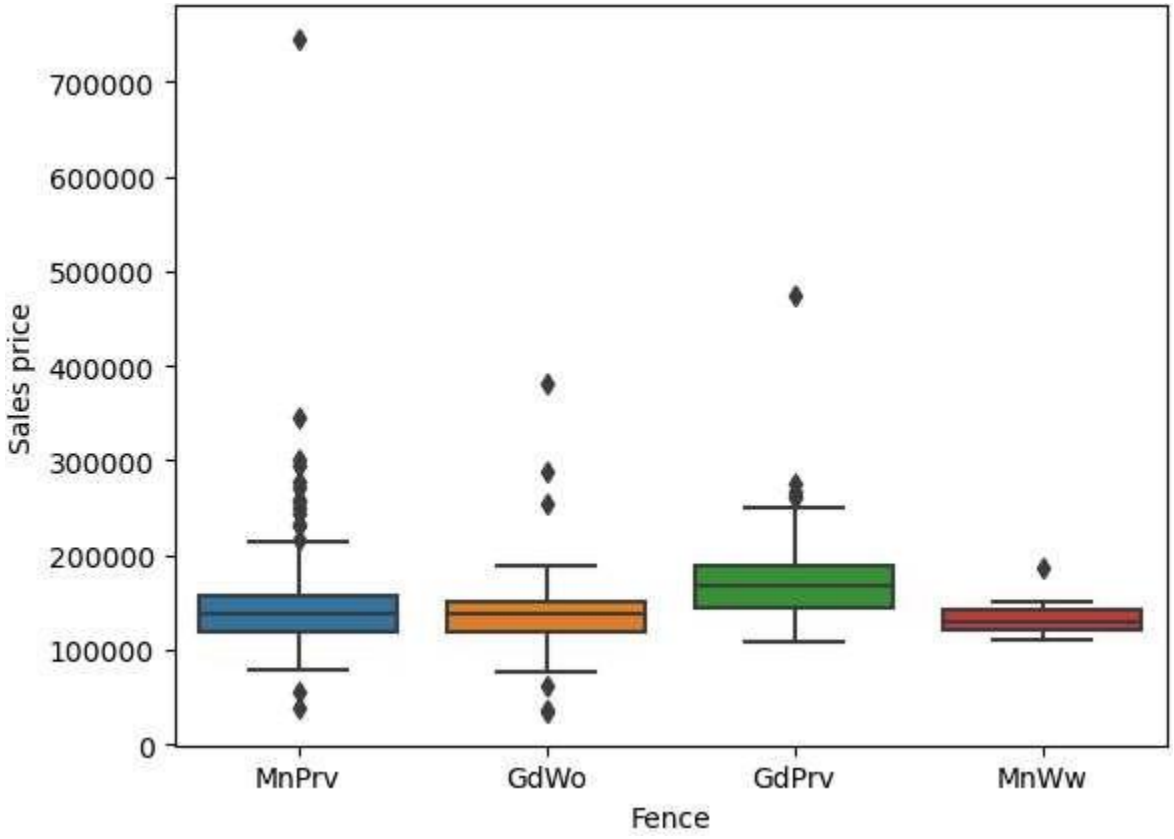

700000
600000
500000
400000
300000
200000
100000
0
Sales price
RFn
Unf
Fin
GarageFinish

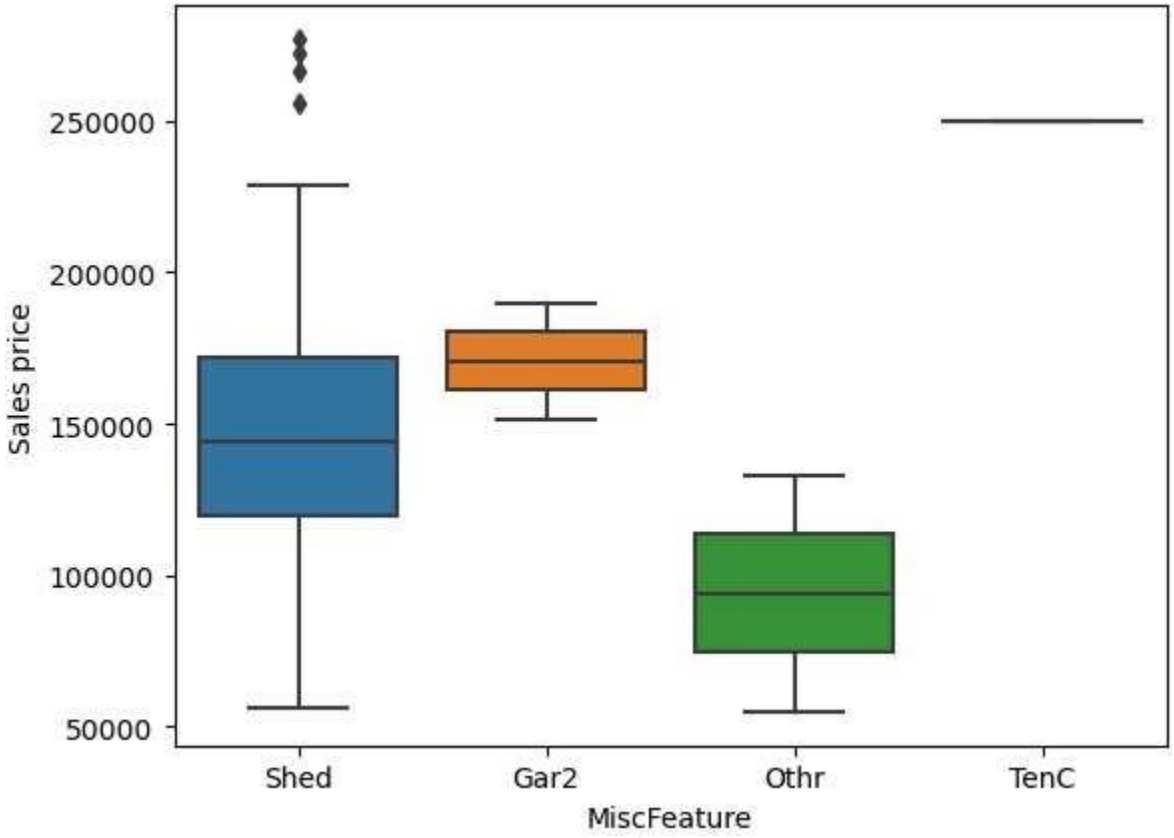

700000
600000
500000
400000
300000
200000
100000
0
Sales price
TA
Fa
Gd
Ex
Po
GarageQual

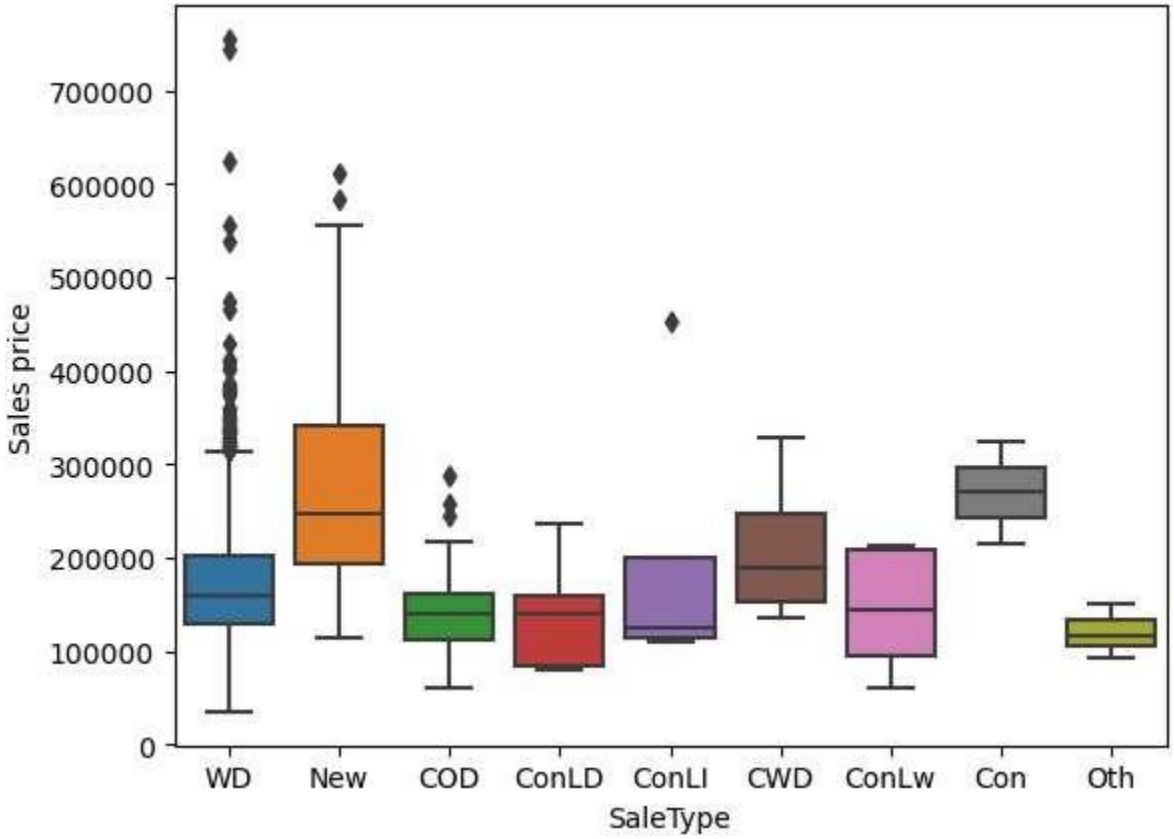

700000
600000
500000
400000
300000
200000
100000
0
Sales price
TA
Fa
Gd
Po
Ex
GarageCond

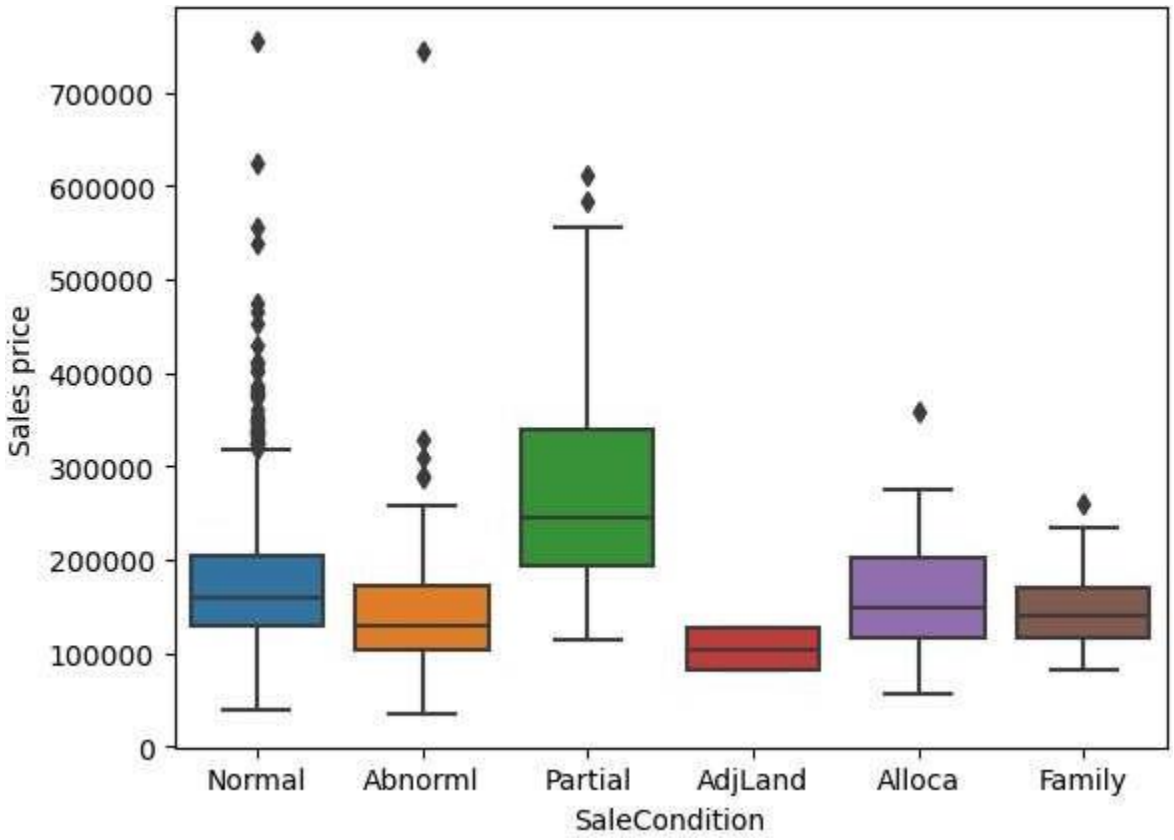

700000
600000
500000
400000
300000
200000
100000
0
Sales price
Y
N
P
PavedDrive


700000
600000
500000
400000
300000
200000
Sales price
Ex
Fa
Gd
PoolQC


700000
600000
500000
400000
300000
200000
100000
0
Sales price
MnPrv
GdWo
GdPrv
MnWw
Fence


250000
200000
150000
100000
50000
Sales price
Shed
Gar2
Othr
TenC
MiscFeature


700000
600000
500000
400000
300000
200000
100000
0
Sales price
WD
New
COD
ConLD
ConLI
CWD
ConLw
Con
Oth
SaleType



Here in the box plot shows that there is a relation between the sale price vs categorical features

Step 4 : separate X and Y (SalePrice) features

In [14]:
```
X = df.drop(columns=['Id','SalePrice'])
Y = df[['SalePrice']]
```

In [15]:
```
X.head()
```

Out[15]:

	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Con
0	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	CollgCr	
1	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	FR2	Gtl	Veenker	
2	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	CollgCr	
3	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	Crawfor	
4	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	FR2	Gtl	NoRidge	

In [16]:
```
Y.head()
```

Out[16]:

	SalePrice
0	208500
1	181500
2	223500
3	140000
4	250000

step 5 : create a preprocessing pipeline for X

In [17]:
```
cat= X.columns[X.dtypes=='object']
con = X.columns[X.dtypes!='object']
```

In [18]:
```
cat
```

Out[18]:
```
Index(['MSZoning', 'Street', 'Alley', 'LotShape', 'LandContour', 'Utilities',
       'LotConfig', 'LandSlope', 'Neighborhood', 'Condition1', 'Condition2',
       'BldgType', 'HouseStyle', 'RoofStyle', 'RoofMatl', 'Exterior1st',
       'Exterior2nd', 'MasVnrType', 'ExterQual', 'ExterCond', 'Foundation',
       'BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2',
       'Heating', 'HeatingQC', 'CentralAir', 'Electrical', 'KitchenQual',
       'Functional', 'FireplaceQu', 'GarageType', 'GarageFinish', 'GarageQual',
       'GarageCond', 'PavedDrive', 'PoolQC', 'Fence', 'MiscFeature',
       'SaleType', 'SaleCondition'],
      dtype='object')
```

In [19]:
```
con
```

Out[19]:
```
Index(['MSSubClass', 'LotFrontage', 'LotArea', 'OverallQual', 'OverallCond',
       'YearBuilt', 'YearRemodAdd', 'MasVnrArea', 'BsmtFinSF1', 'BsmtFinSF2',
       'BsmtUnfSF', 'TotalBsmtSF', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF',
       'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath', 'HalfBath',
       'BedroomAbvGr', 'KitchenAbvGr', 'TotRmsAbvGrd', 'Fireplaces',
       'GarageYrBlt', 'GarageCars', 'GarageArea', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'MiscVal',
       'MoSold', 'YrSold'],
      dtype='object')
```

In [20]:
```
from sklearn.pipeline import Pipeline
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler,OrdinalEncoder
from sklearn.compose import ColumnTransformer
```

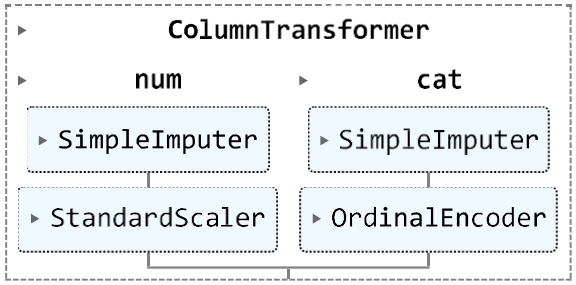
In [21]:
```
num_pipe = Pipeline(steps=[('impute',SimpleImputer(strategy='median')),
                           ('scaler', StandardScaler())])
```

In [22]:
```
cat_pipe = Pipeline(steps=[('impute',SimpleImputer(strategy='constant', fill_value='Not_Avail')),
                           ('ohe',OrdinalEncoder())])
```

In [23]:
```
pre = ColumnTransformer([('num', num_pipe, con),
                         ('cat',cat_pipe,cat)]).set_output(transform='pandas')
```

In [24]:
```
pre
```

Out[24]:



In [25]:
```
X_pre = pre.fit_transform(X)
X_pre.head()
```

Out[25]:

	num__MSSubClass	num__LotFrontage	num__LotArea	num__OverallQual	num__OverallCond	num__YearBuilt	num__YearRemodAdd	num__M
0	0.073375	-0.220875	-0.207142	0.651479	-0.517200	1.050994	0.878668	
1	-0.872563	0.460320	-0.091886	-0.071836	2.179628	0.156734	-0.429577	
2	0.073375	-0.084636	0.073480	0.651479	-0.517200	0.984752	0.830215	
3	0.309859	-0.447940	-0.096897	0.651479	-0.517200	-1.863632	-0.720298	
4	0.073375	0.641972	0.375148	1.374795	-0.517200	0.951632	0.733308	

step 6 : Backward feature selection

In [26]:
```
# feature selection
from sklearn.linear_model import LinearRegression
from sklearn.feature_selection import SequentialFeatureSelector
lr = LinearRegression()
sel = SequentialFeatureSelector(lr,n_features_to_select='auto',direction='backward')
sel.fit(X_pre,Y)
sel_cols = sel.get_feature_names_out()
sel_cols
```

Out[26]:
```
array(['num__MSSubClass', 'num__LotArea', 'num__OverallQual',
       'num__OverallCond', 'num__YearBuilt', 'num__MasVnrArea',
       'num__1stFlrSF', 'num__2ndFlrSF', 'num__LowQualFinSF',
       'num__GrLivArea', 'num__BsmtFullBath', 'num__KitchenAbvGr',
       'num__TotRmsAbvGrd', 'num__Fireplaces', 'num__GarageCars',
       'num__WoodDeckSF', 'num__ScreenPorch', 'num__PoolArea',
       'num__YrSold', 'cat__LandContour', 'cat__Neighborhood',
       'cat__BldgType', 'cat__HouseStyle', 'cat__RoofMatl',
       'cat__Exterior1st', 'cat__MasVnrType', 'cat__ExterQual',
       'cat__ExterCond', 'cat__Foundation', 'cat__BsmtQual',
       'cat__BsmtCond', 'cat__BsmtExposure', 'cat__BsmtFinType1',
       'cat__BsmtFinType2', 'cat__KitchenQual', 'cat__Functional',
       'cat__GarageFinish', 'cat__PavedDrive', 'cat__MiscFeature',
       'cat__SaleCondition'], dtype=object)
```

In [27]:
```
len(sel_cols)
```

Out[27]:
```
40
```

In [28]:
```
sel_cols[0]
```

Out[28]:
```
'num__MSSubClass'
```

In [29]:
```
sel_cols[0].split('__')
```

Out[29]:
```
['num', 'MSSubClass']
```

In [30]:
```
sel_cols[0].split('__')[1]
```

Out[30]:
```
'MSSubClass'
```

In [31]:
```
# Extracting original column name
imp_cols = []
for i in sel_cols:
    s = i.split('__')[1]
    imp_cols.append(s)
```

In [32]:
```
imp_cols
```

Out[32]:
```
['MSSubClass',
 'LotArea',
 'OverallQual',
 'OverallCond',
 'YearBuilt',
 'MasVnrArea',
 '1stFlrSF',
 '2ndFlrSF',
 'LowQualFinSF',
 'GrLivArea',
 'BsmtFullBath',
 'KitchenAbvGr',
 'TotRmsAbvGrd',
 'Fireplaces',
 'GarageCars',
 'WoodDeckSF',
 'ScreenPorch',
 'PoolArea',
 'YrSold',
 'LandContour',
 'Neighborhood',
 'BldgType',
 'HouseStyle',
 'RoofMatl',
 'Exterior1st',
 'MasVnrType',
 'ExterQual',
 'ExterCond',
 'Foundation',
 'BsmtQual',
 'BsmtCond',
 'BsmtExposure',
 'BsmtFinType1',
 'BsmtFinType2',
 'KitchenQual',
 'Functional',
 'GarageFinish',
 'PavedDrive',
 'MiscFeature',
 'SaleCondition']
```

In [33]:
```
X_sel = X[imp_cols]
X_sel
```

Out[33]:

	MSSubClass	LotArea	OverallQual	OverallCond	YearBuilt	MasVnrArea	1stFlrSF	2ndFlrSF	LowQualFinSF	GrLivArea	BsmtFullBath	Ki
0	60	8450	7	5	2003	196.0	856	854	0	1710	1	
1	20	9600	6	8	1976	0.0	1262	0	0	1262	0	
2	60	11250	7	5	2001	162.0	920	866	0	1786	1	
3	70	9550	7	5	1915	0.0	961	756	0	1717	1	
4	60	14260	8	5	2000	350.0	1145	1053	0	2198	1	
...	...	...	...	...	...	...	...	...	...	...	...	
1455	60	7917	6	5	1999	0.0	953	694	0	1647	0	
1456	20	13175	6	6	1978	119.0	2073	0	0	2073	1	
1457	70	9042	7	9	1941	0.0	1188	1152	0	2340	0	
1458	20	9717	5	6	1950	0.0	1078	0	0	1078	1	
1459	20	9937	5	6	1965	0.0	1256	0	0	1256	1	

1460 rows × 40 columns

Step 7 : Create a final pipeline

categorical - OneHotEncoder

In [34]:
```
cat_sel = list(X_sel.columns[X_sel.dtypes=='object'])
con_sel = list(X_sel.columns[X_sel.dtypes!='object'])
```

In [35]:
```
cat_sel
```

Out[35]:
```
['LandContour',
 'Neighborhood',
 'BldgType',
 'HouseStyle',
 'RoofMatl',
 'Exterior1st',
 'MasVnrType',
 'ExterQual',
 'ExterCond',
 'Foundation',
 'BsmtQual',
 'BsmtCond',
 'BsmtExposure',
 'BsmtFinType1',
 'BsmtFinType2',
 'KitchenQual',
 'Functional',
 'GarageFinish',
 'PavedDrive',
 'MiscFeature',
 'SaleCondition']
```

In [36]:
```
con_sel
```

Out[36]:
```
['MSSubClass',
 'LotArea',
 'OverallQual',
 'OverallCond',
 'YearBuilt',
 'MasVnrArea',
 '1stFlrSF',
 '2ndFlrSF',
 'LowQualFinSF',
 'GrLivArea',
 'BsmtFullBath',
 'KitchenAbvGr',
 'TotRmsAbvGrd',
 'Fireplaces',
 'GarageCars',
 'WoodDeckSF',
 'ScreenPorch',
 'PoolArea',
 'YrSold']
```

In [37]:
```
from sklearn.preprocessing import OneHotEncoder
```

In [38]:
```
num_pipe1 = Pipeline(steps=[('impute',SimpleImputer(strategy='median')),
                            ('scaler',StandardScaler())])
```

In [39]:
```
cat_pipe1 = Pipeline([('impute',SimpleImputer(strategy='constant', fill_value='Not_Avail')),
                      ('ohe',OneHotEncoder(handle_unknown='ignore',sparse_output=False))])
```

In [40]:
```
pre1 = ColumnTransformer([('num',num_pipe1,con_sel),
                          ('cat',cat_pipe1,cat_sel)]).set_output(transform='pandas')
```

In [41]:
```
X_sel_pre = pre1.fit_transform(X_sel)
X_sel_pre.head()
```

Out[41]:

	num__MSSubClass	num__LotArea	num__OverallQual	num__OverallCond	num__YearBuilt	num__MasVnrArea	num__1stFlrSF	num__2ndFlrSF
0	0.073375	-0.207142	0.651479	-0.517200	1.050994	0.514104	-0.793434	1.161852
1	-0.872563	-0.091886	-0.071836	2.179628	0.156734	-0.570750	0.257140	-0.795163
2	0.073375	0.073480	0.651479	-0.517200	0.984752	0.325915	-0.627826	1.189351
3	0.309859	-0.096897	0.651479	-0.517200	-1.863632	-0.570750	-0.521734	0.937276
4	0.073375	0.375148	1.374795	-0.517200	0.951632	1.366489	-0.045611	1.617877

Step 8: Train Test split

20% Unseen to model

In [42]:
```
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest = train_test_split(X_sel_pre,Y,test_size=0.20,random_state=42)
```

In [43]:
```
xtrain.head()
```

Out[43]:

	num__MSSubClass	num__LotArea	num__OverallQual	num__OverallCond	num__YearBuilt	num__MasVnrArea	num__1stFlrSF	num__2ndFlr
254	-0.872563	-0.212153	-0.795151	0.381743	-0.472560	-0.570750	0.391697	-0.7951
1066	0.073375	-0.268578	-0.071836	1.280685	0.719786	-0.570750	-0.940928	0.9739
638	-0.636078	-0.174369	-0.795151	1.280685	-2.029235	-0.570750	-0.948691	-0.7951
799	-0.163109	-0.332419	-0.795151	1.280685	-1.134975	0.824062	-0.469981	1.0083
380	-0.163109	-0.552908	-0.795151	0.381743	-1.565545	-0.570750	-0.353538	0.7287

In [44]:
```
ytrain.head()
```

Out[44]:

	SalePrice
254	145000
1066	178000
638	85000
799	175000
380	127000

In [45]:
```
xtest.head()
```

Out[45]:

	num__MSSubClass	num__LotArea	num__OverallQual	num__OverallCond	num__YearBuilt	num__MasVnrArea	num__1stFlrSF	num__2ndFlr
892	-0.872563	-0.210750	-0.071836	2.179628	-0.273836	-0.570750	-0.244858	-0.7951
1105	0.073375	0.174303	1.374795	-0.517200	0.752907	1.432909	0.872994	1.7759
413	-0.636078	-0.156028	-0.795151	0.381743	-1.466183	-0.570750	-0.348363	-0.7951
522	-0.163109	-0.552908	-0.071836	1.280685	-0.803768	-0.570750	-0.410466	0.7172
1036	-0.872563	0.238646	2.098110	-0.517200	1.183477	-0.183302	1.183509	-0.7951

In [46]:
```
ytest.head()
```

Out[46]:

	SalePrice
892	154500
1105	325000
413	115000
522	159000
1036	315500

Step 9 : Building Model and Evaluate Model

In [47]:
```
model_linear = LinearRegression()
model_linear.fit(xtrain,ytrain)
```

Out[47]:
```
LinearRegression()
```

In [48]:
```
model_linear.score(xtrain,ytrain)
```

Out[48]:
```
0.9076692918212752
```

In [49]:
```
model_linear.score(xtest,ytest)
```

Out[49]:
```
-9.329313458976195e+18
```

In [50]:
```
def adj_r2(model,xtrain,ytrain):
    r2 = model.score(xtrain,ytrain)
    N = xtrain.shape[0]
    p = xtrain.shape[1]
    num = (1 - r2)*(N - 1)
    den = N-p-1
    r2a = 1 - (num/den)
    return r2a
```

In [51]:
```
adj_r2(model_linear,xtrain,ytrain)
```

Out[51]:
```
0.8928927073115588
```

In [52]:
```
from sklearn.metrics import mean_squared_error,mean_absolute_error,r2_score
def evaluate_model(model_linear,x,y):
    ypred = model_linear.predict(x)
    mse = mean_squared_error(y, ypred)
    rmse = (mse)**(1/2)
    mae = mean_absolute_error(y, ypred)
    r2 = r2_score(y, ypred)
    print(f'Mean squared error : {mse:.2f}')
    print(f'Root Mean Squared Error : {rmse:.2f}')
    print(f'Mean absolute error : {mae:.2f}')
    print(f'R2 Score : {r2:.4f}')
```

In [53]:
```
evaluate_model(model_linear,xtrain,ytrain)
```

```
Mean squared error : 550711160.53
Root Mean Squared Error : 23467.24
Mean absolute error : 14703.85
R2 Score : 0.9077
```

In [54]:
```
evaluate_model(model_linear,xtest,ytest)
```

```
Mean squared error : 71558877519448675591254966272.00
Root Mean Squared Error : 267504911206221.94
Mean absolute error : 22957694616873.65
R2 Score : -93293134589761945600.0000
```

step 10 : Ridge Model

In [55]:
```
from sklearn.linear_model import Ridge
model_ridge = Ridge(alpha=2)
model_ridge.fit(xtrain,ytrain)
```

Out[55]:
```
Ridge(alpha=2)
```

In [56]:
```
model_ridge.score(xtrain,ytrain)
```

Out[56]:
```
0.8943247680090605
```

In [57]:
```
model_ridge.score(xtest,ytest)
```

Out[57]:
```
0.8853384288073846
```

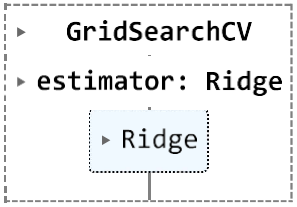
Tuning the ridge model with GridSearchCV

In [58]:
```
import numpy as np
params = {'alpha':np.arange(start=1,stop=100,step=1)}
params
```

Out[58]:
```
{'alpha': array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
        18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
        35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51,
        52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68,
        69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85,
        86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99])}
```

In [59]:
```
from sklearn.model_selection import GridSearchCV
rr = Ridge()
gscv = GridSearchCV(rr, param_grid=params, cv=5,scoring='neg_mean_squared_error')
gscv.fit(xtrain,ytrain)
```

Out[59]:



In [60]:
```
gscv.best_params_
```

Out[60]:
```
{'alpha': 2}
```

In [61]:
```
gscv.best_score_
```

Out[61]:
```
-1021385798.4484398
```

In [62]:
```
best_ridge = gscv.best_estimator_
best_ridge
```

Out[62]:
```
Ridge
Ridge(alpha=2)
```

In [63]:
```
best_ridge.score(xtrain,ytrain)
```

Out[63]:
```
0.8943247680090605
```

In [64]:
```
best_ridge.score(xtest,ytest)
```

Out[64]:
```
0.8853384288073846
```

In [65]:
```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
def evaluate_model(model, x, y):
    ypred = model.predict(x)
    mse = mean_squared_error(y, ypred)
    rmse = mse**(1/2)
    mae = mean_absolute_error(y, ypred)
    r2 = r2_score(y, ypred)
    print(f'Mean Squared Error : {mse:.2f}')
    print(f'Root Mean Squared Error : {rmse:.2f}')
    print(f'Mean Absolute Error : {mae:.2f}')
    print(f'R2 Score : {r2:.4f}')
```

In [66]:
```
evaluate_model(best_ridge,xtrain,ytrain)
```

```
Mean Squared Error : 630305245.10
Root Mean Squared Error : 25105.88
Mean Absolute Error : 15453.79
R2 Score : 0.8943
```

In [67]:
```
evaluate_model(best_ridge,xtest,ytest)
```

```
Mean Squared Error : 879491654.48
Root Mean Squared Error : 29656.22
Mean Absolute Error : 18428.29
R2 Score : 0.8853
```

In [68]:
```
from sklearn.model_selection import cross_val_score
ridge_score = cross_val_score(best_ridge,xtrain,ytrain,cv=5,scoring='r2')
ridge_score
```

Out[68]:
```
array([0.83948567, 0.79133989, 0.73936147, 0.89702828, 0.90793293])
```

In [69]:
```
ridge_score.mean()
```

Out[69]:
```
0.8350296474691161
```

step 11 : Lasso Model

In [70]:
```
from sklearn.linear_model import Lasso
model_lasso = Lasso(alpha=1)
model_lasso.fit(xtrain,ytrain)
```

Out[70]:
```
Lasso(alpha=1)
```

In [71]:
```
model_lasso.score(xtrain,ytrain)
```

Out[71]:
```
0.9076721784324515
```

In [72]:
```
model_lasso.score(xtest,ytest)
```

Out[72]:
```
0.8965579093790569
```

In [73]:
```
params
```

Out[73]:
```
{'alpha': array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16, 17,
        18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34,
        35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51,
        52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68,
        69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85,
        86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99])}
```

In [74]:
```
ls = Lasso()
gscv2 = GridSearchCV(ls,param_grid=params, cv=5, scoring='neg_mean_squared_error')
gscv2.fit(xtrain,ytrain)
```

Out[74]:

GridSearchCV
estimator: Lasso
Lasso

In [75]:
```
gscv.best_params_
```

Out[75]:
```
{'alpha': 2}
```

In [76]:
```
gscv.best_score_
```

Out[76]:
```
-1021385798.4484398
```

In [77]:
```
best_lasso = gscv.best_estimator_
best_lasso
```

Out[77]:

Ridge
Ridge(alpha=2)

In [78]:
```
best_lasso.score(xtrain,ytrain)
```

Out[78]:
```
0.8943247680090605
```

In [79]:
```
best_lasso.score(xtest,ytest)
```

Out[79]:
```
0.8853384288073846
```

In [80]:
```
evaluate_model(best_lasso,xtrain,ytrain)
```

```
Mean Squared Error : 630305245.10
Root Mean Squared Error : 25105.88
Mean Absolute Error : 15453.79
R2 Score : 0.8943
```

In [81]:
```
evaluate_model(best_lasso,xtest,ytest)
```

```
Mean Squared Error : 879491654.48
Root Mean Squared Error : 29656.22
Mean Absolute Error : 18428.29
R2 Score : 0.8853
```

In [82]:
```
scores_lasso = cross_val_score(best_lasso, xtrain, ytrain, cv=5, scoring='r2')
scores_lasso
```

Out[82]:
```
array([0.83948567, 0.79133989, 0.73936147, 0.89702828, 0.90793293])
```

In [83]:
```
scores_lasso.mean()
```

Out[83]:
```
0.8350296474691161
```

In [84]:
```
ypred_train = best_lasso.predict(xtrain)
ypred_test = best_lasso.predict(xtest)
```

In [85]:
```
ypred_train[0:5]
```

Out[85]:
```
array([[141181.41808509],
       [178149.6881175 ],
       [ 93525.44790506],
       [174373.44657843],
       [158426.04182418]])
```

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

Out[86]:

	SalePrice
254	145000
1066	178000
638	85000
799	175000
380	127000

In [87]:
```
ypred_test[0:5]
```

Out[87]:
```
array([[150019.67689962],
       [348376.85414676],
       [108048.20077915],
       [183210.39233151],
       [332701.37511536]])
```

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

Out[88]:

	SalePrice
892	154500
1105	325000
413	115000
522	159000
1036	315500

Step 12 : Predicting out of sample data

In [89]:
```
xnew = pd.read_csv('sample_set.csv')
xnew
```

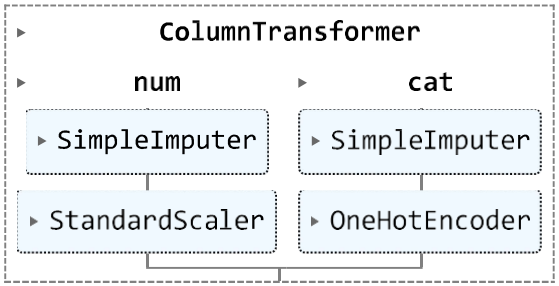
Out[89]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborh
0	1461	20	RH	80.0	11622	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	NA
1	1462	20	RL	81.0	14267	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	NA
2	1463	60	RL	74.0	13830	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	Gi
3	1464	60	RL	78.0	9978	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	Gi
4	1465	120	RL	43.0	5005	Pave	NaN	IR1	HLS	AllPub	Inside	Gtl	Sto
...	...	...	...	...	...	...	...	...	...	...	...	...	
1454	2915	160	RM	21.0	1936	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Mead
1455	2916	160	RM	21.0	1894	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Mead
1456	2917	20	RL	160.0	20000	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Mit
1457	2918	85	RL	62.0	10441	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Mit
1458	2919	60	RL	74.0	9627	Pave	NaN	Reg	Lvl	AllPub	Inside	Mod	Mit

1459 rows × 80 columns

In [90]:
```
pre1
```

Out[90]:



In [91]:
```
xnew_pre = pre1.transform(xnew)
xnew_pre
```

Out[91]:

	num__MSSubClass	num__LotArea	num__OverallQual	num__OverallCond	num__YearBuilt	num__MasVnrArea	num__1stFlrSF	num__2ndFlr
0	-0.872563	0.110763	-0.795151	0.381743	-0.340077	-0.570750	-0.689929	-0.7951
1	-0.872563	0.375850	-0.071836	0.381743	-0.439440	0.027027	0.430511	-0.7951
2	0.073375	0.332053	-0.795151	-0.517200	0.852269	-0.570750	-0.607125	0.8112
3	0.073375	-0.054002	-0.071836	0.381743	0.885390	-0.460051	-0.612300	0.7585
4	1.492282	-0.552407	1.374795	-0.517200	0.686666	-0.570750	0.303718	-0.7951
...	...	...	...	...	...	...	...	
1454	2.438219	-0.859988	-1.518467	1.280685	-0.041991	-0.570750	-1.595596	0.4560
1455	2.438219	-0.864197	-1.518467	-0.517200	-0.041991	-0.570750	-1.595596	0.4560
1456	-0.872563	0.950423	-0.795151	1.280685	-0.373198	-0.570750	0.158811	-0.7951
1457	0.664586	-0.007600	-0.795151	-0.517200	0.686666	-0.570750	-0.498445	-0.7951
1458	0.073375	-0.089180	0.651479	-0.517200	0.719786	-0.050463	-0.431167	1.5055

1459 rows × 161 columns

In [92]:
```
preds = best_lasso.predict(xnew_pre)
preds
```

Out[92]:
```
array([[115079.22175753],
       [147011.47227897],
       [172008.80698563],
       ...,
       [166010.21300068],
       [118056.0487289 ],
       [209366.27917646]])
```

final step : Save the above in dataframe and in csv format

In [93]:
```
xnew['SalePrice_pred'] = preds
```

In [94]:
```
xnew
```

Out[94]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborh
0	1461	20	RH	80.0	11622	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	NA
1	1462	20	RL	81.0	14267	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	NA
2	1463	60	RL	74.0	13830	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	Gi
3	1464	60	RL	78.0	9978	Pave	NaN	IR1	Lvl	AllPub	Inside	Gtl	Gi
4	1465	120	RL	43.0	5005	Pave	NaN	IR1	HLS	AllPub	Inside	Gtl	Sto
...	...	...	...	...	...	...	...	...	...	...	...	...	
1454	2915	160	RM	21.0	1936	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Mead
1455	2916	160	RM	21.0	1894	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Mead
1456	2917	20	RL	160.0	20000	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Mit
1457	2918	85	RL	62.0	10441	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Mit
1458	2919	60	RL	74.0	9627	Pave	NaN	Reg	Lvl	AllPub	Inside	Mod	Mit

1459 rows × 81 columns

In [95]:
```
xnew[['SalePrice_pred']].to_csv('House_preds_Result.csv')
```