```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import scipy.stats as stats


#Import and suppress warnings
import warnings
warnings.filterwarnings('ignore')


df = pd.read_csv("PEP1.csv") #csv to df
display(df.head(10))
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl
5	6	50	RL	85.0	14115	Pave	NaN	IR1	Lvl
6	7	20	RL	75.0	10084	Pave	NaN	Reg	Lvl
7	8	60	RL	NaN	10382	Pave	NaN	IR1	Lvl
8	9	50	RM	51.0	6120	Pave	NaN	Reg	Lvl
9	10	190	RL	50.0	7420	Pave	NaN	Reg	Lvl

10 rows × 81 columns

```
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     1452 non-null   object
26  MasVnrArea     1452 non-null   float64
27  ExterQual      1460 non-null   object
28  ExterCond      1460 non-null   object
29  Foundation     1460 non-null   object
30  BsmtQual       1423 non-null   object
31  BsmtCond       1423 non-null   object
32  BsmtExposure   1422 non-null   object
33  BsmtFinType1   1423 non-null   object
34  BsmtFinSF1     1460 non-null   int64
35  BsmtFinType2   1422 non-null   object
36  BsmtFinSF2     1460 non-null   int64
37  BsmtUnfSF      1460 non-null   int64
38  TotalBsmtSF    1460 non-null   int64
39  Heating        1460 non-null   object
40  HeatingQC      1460 non-null   object
41  CentralAir     1460 non-null   object
42  Electrical     1459 non-null   object
43  1stFlrSF       1460 non-null   int64
44  2ndFlrSF       1460 non-null   int64
45  LowQualFinSF   1460 non-null   int64
46  GrLivArea      1460 non-null   int64
47  BsmtFullBath   1460 non-null   int64
48  BsmtHalfBath   1460 non-null   int64
49  FullBath       1460 non-null   int64
50  HalfBath       1460 non-null   int64
51  BedroomAbvGr   1460 non-null   int64
52  KitchebvGr     1460 non-null   int64
```

Task 1 - Understanding the Dataset a. Identify the shape of the dataset b. Identify variables with null values c. Identify variables with unique values

```
df.size
```

```
118260
```

```
df.columns
```

```
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', 'KitchebvGr', 'KitchenQual', 'TotRmsAbvGrd',
       'Functiol', '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')
```

```
df.shape
```

```
(1460, 81)
```

```
df.ndim
```

```
2
```

```
df.isna().sum()
```

Show hidden output

```
#print dataframe
for i in df.columns:
    print (i, df[i].unique())
    print()
```

Show hidden output

```
print((df.count(), len(df.columns)))
```

Show hidden output

```
df.count()
```

```
Id               1460
MSSubClass       1460
MSZoning         1460
LotFrontage      1201
```

```
LotArea          1460
                 ...
MoSold           1460
YrSold           1460
SaleType         1460
SaleCondition    1460
SalePrice        1460
Length: 81, dtype: int64
```

Task 2 - Selecting the numerical and categorical variables

```
numeric_df= df.select_dtypes(include=[np.number])
category_df=df.select_dtypes(exclude=[np.number])
```

```
numeric_df.columns
```

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

```
category_df.columns
```

```
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',
       'Functiol', 'FireplaceQu', 'GarageType', 'GarageFinish', 'GarageQual',
       'GarageCond', 'PavedDrive', 'PoolQC', 'Fence', 'MiscFeature',
       'SaleType', 'SaleCondition'],
      dtype='object')
```

Task 3 & 4 - Descriptive Stats and EDA Numerical & Categorical

```
#Analyzing numerical variables
df.describe().T
```

	count	mean	std	min	25%	50%	7
Id	1460.0	730.500000	421.610009	1.0	365.75	730.5	1095.
MSSubClass	1460.0	56.897260	42.300571	20.0	20.00	50.0	70.(
LotFrontage	1201.0	70.049958	24.284752	21.0	59.00	69.0	80.(
LotArea	1460.0	10516.828082	9981.264932	1300.0	7553.50	9478.5	11601.
OverallQual	1460.0	6.099315	1.382997	1.0	5.00	6.0	7.(
OverallCond	1460.0	5.575342	1.112799	1.0	5.00	5.0	6.(
YearBuilt	1460.0	1971.267808	30.202904	1872.0	1954.00	1973.0	2000.(
YearRemodAdd	1460.0	1984.865753	20.645407	1950.0	1967.00	1994.0	2004.(
MasVnrArea	1452.0	103.685262	181.066207	0.0	0.00	0.0	166.(
BsmtFinSF1	1460.0	443.639726	456.098091	0.0	0.00	383.5	712.
BsmtFinSF2	1460.0	46.549315	161.319273	0.0	0.00	0.0	0.(
BsmtUnfSF	1460.0	567.240411	441.866955	0.0	223.00	477.5	808.(
TotalBsmtSF	1460.0	1057.429452	438.705324	0.0	795.75	991.5	1298.
1stFlrSF	1460.0	1162.626712	386.587738	334.0	882.00	1087.0	1391.
2ndFlrSF	1460.0	346.992466	436.528436	0.0	0.00	0.0	728.(
LowQualFinSF	1460.0	5.844521	48.623081	0.0	0.00	0.0	0.(
GrLivArea	1460.0	1515.463699	525.480383	334.0	1129.50	1464.0	1776.
BsmtFullBath	1460.0	0.425342	0.518911	0.0	0.00	0.0	1.(
BsmtHalfBath	1460.0	0.057534	0.238753	0.0	0.00	0.0	0.(
FullBath	1460.0	1.565068	0.550916	0.0	1.00	2.0	2.(
HalfBath	1460.0	0.382877	0.502885	0.0	0.00	0.0	1.(
BedroomAbvGr	1460.0	2.866438	0.815778	0.0	2.00	3.0	3.(
KitchebvGr	1460.0	1.046575	0.220338	0.0	1.00	1.0	1.(
TotRmsAbvGrd	1460.0	6.517808	1.625393	2.0	5.00	6.0	7.(
Fireplaces	1460.0	0.613014	0.644666	0.0	0.00	1.0	1.(
GarageYrBlt	1379.0	1978.506164	24.689725	1900.0	1961.00	1980.0	2002.(
GarageCars	1460.0	1.767123	0.747315	0.0	1.00	2.0	2.(
GarageArea	1460.0	472.980137	213.804841	0.0	334.50	480.0	576.(
WoodDeckSF	1460.0	94.244521	125.338794	0.0	0.00	0.0	168.(
OpenPorchSF	1460.0	46.660274	66.256028	0.0	0.00	25.0	68.(

EnclosedPorch	1460.0	21.954110	61.119149	0.0	0.00	0.0	0.(
3SsnPorch	1460.0	3.409589	29.317331	0.0	0.00	0.0	0.(
ScreenPorch	1460.0	15.060959	55.757415	0.0	0.00	0.0	0.(
PoolArea	1460.0	2.758904	40.177307	0.0	0.00	0.0	0.(
MiscVal	1460.0	43.489041	496.123024	0.0	0.00	0.0	0.(
MoSold	1460.0	6.321918	2.703626	1.0	5.00	6.0	8.(

```
#Missing Value Detection
df.isna().any()

# check how much data is missing
df.isnull().sum()
```

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

```
# Checking the skewness of entire data
df.skew()
```

```
Id               0.000000
MSSubClass       1.407657
LotFrontage      2.163569
LotArea         12.207688
OverallQual      0.216944
OverallCond      0.693067
YearBuilt       -0.613461
YearRemodAdd    -0.503562
MasVnrArea       2.669084
BsmtFinSF1       1.685503
BsmtFinSF2       4.255261
BsmtUnfSF        0.920268
TotalBsmtSF      1.524255
1stFlrSF         1.376757
2ndFlrSF         0.813030
LowQualFinSF     9.011341
GrLivArea        1.366560
BsmtFullBath     0.596067
BsmtHalfBath     4.103403
FullBath         0.036562
HalfBath         0.675897
BedroomAbvGr     0.211790
```

```
KitchebvGr        4.488397
TotRmsAbvGrd      0.676341
Fireplaces        0.649565
GarageYrBlt      -0.649415
GarageCars       -0.342549
GarageArea        0.179981
WoodDeckSF        1.541376
OpenPorchSF       2.364342
EnclosedPorch     3.089872
3SsnPorch        10.304342
ScreenPorch       4.122214
PoolArea         14.828374
MiscVal          24.476794
MoSold            0.212053
YrSold            0.096269
SalePrice         1.882876
dtype: float64
```

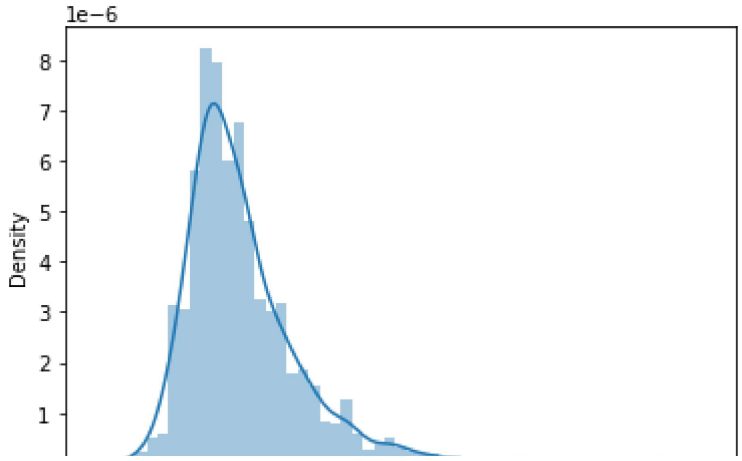
```
# Checking skewness and kurtosisof SalePrice
#df["SalePrice"].skew()
print("Skewness: %f" % df['SalePrice'].skew())
print("Kurtosis: %f" % df['SalePrice'].kurt())
```

```
Skewness: 1.882876
Kurtosis: 6.536282
```

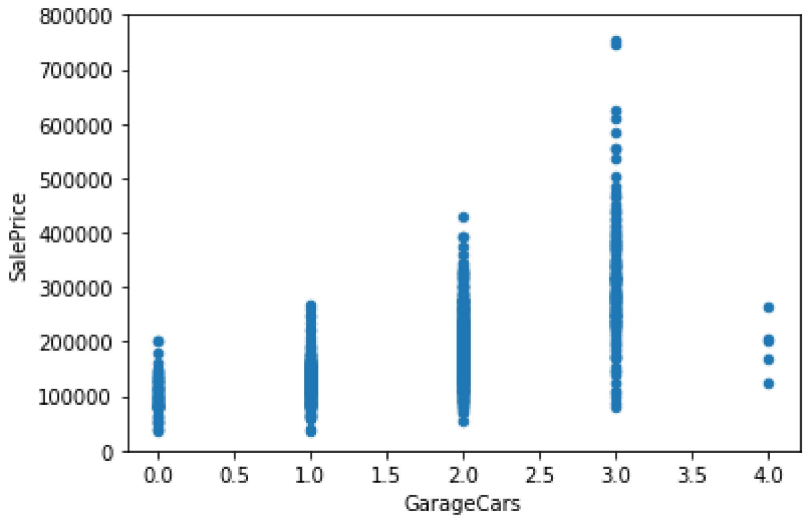
```
#Find correlation
df.corr()
```

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond
Id	1.000000	0.011156	-0.010601	-0.033226	-0.028365	0.012609
MSSubClass	0.011156	1.000000	-0.386347	-0.139781	0.032628	-0.059316
LotFrontage	-0.010601	-0.386347	1.000000	0.426095	0.251646	-0.059213
LotArea	-0.033226	-0.139781	0.426095	1.000000	0.105806	-0.005636
OverallQual	-0.028365	0.032628	0.251646	0.105806	1.000000	-0.091932
OverallCond	0.012609	-0.059316	-0.059213	-0.005636	-0.091932	1.000000
YearBuilt	-0.012713	0.027850	0.123349	0.014228	0.572323	-0.375983
YearRemodAdd	-0.021998	0.040581	0.088866	0.013788	0.550684	0.073741
MasVnrArea	-0.050298	0.022936	0.193458	0.104160	0.411876	-0.128101
BsmtFinSF1	-0.005024	-0.069836	0.233633	0.214103	0.239666	-0.046231
BsmtFinSF2	-0.005968	-0.065649	0.049900	0.111170	-0.059119	0.040229
BsmtUnfSF	-0.007940	-0.140759	0.132644	-0.002618	0.308159	-0.136841
TotalBsmtSF	-0.015415	-0.238518	0.392075	0.260833	0.537808	-0.171098
1stFlrSF	0.010496	-0.251758	0.457181	0.299475	0.476224	-0.144203
2ndFlrSF	0.005590	0.307886	0.080177	0.050986	0.295493	0.028942
LowQualFinSF	-0.044230	0.046474	0.038469	0.004779	-0.030429	0.025494
GrLivArea	0.008273	0.074853	0.402797	0.263116	0.593007	-0.079686
BsmtFullBath	0.002289	0.003491	0.100949	0.158155	0.111098	-0.054942
BsmtHalfBath	-0.020155	-0.002333	-0.007234	0.048046	-0.040150	0.117821
FullBath	0.005587	0.131608	0.198769	0.126031	0.550600	-0.194149
HalfBath	0.006784	0.177354	0.053532	0.014259	0.273458	-0.060769
BedroomAbvGr	0.037719	-0.023438	0.263170	0.119690	0.101676	0.012980
KitchebvGr	0.002951	0.281721	-0.006069	-0.017784	-0.183882	-0.087001
TotRmsAbvGrd	0.027239	0.040380	0.352096	0.190015	0.427452	-0.057583
Fireplaces	-0.019772	-0.045569	0.266639	0.271364	0.396765	-0.023820
GarageYrBlt	0.000072	0.085072	0.070250	-0.024947	0.547766	-0.324297

```
#histogram
sns.distplot(df['SalePrice']);
```

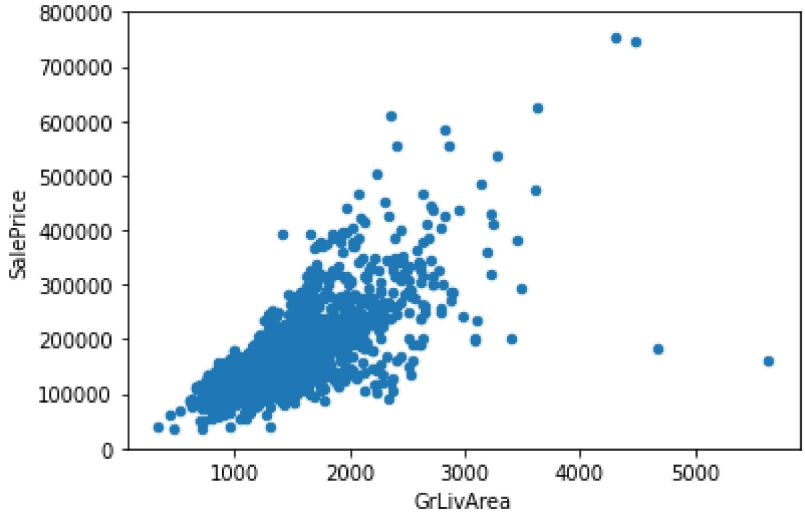



```
#Relationship with numerical variables by scatter plot GarageCars/saleprice
var = 'GarageCars'
data = pd.concat([df['SalePrice'], df[var]], axis=1)
data.plot.scatter(x=var, y='SalePrice', ylim=(0,800000));
```

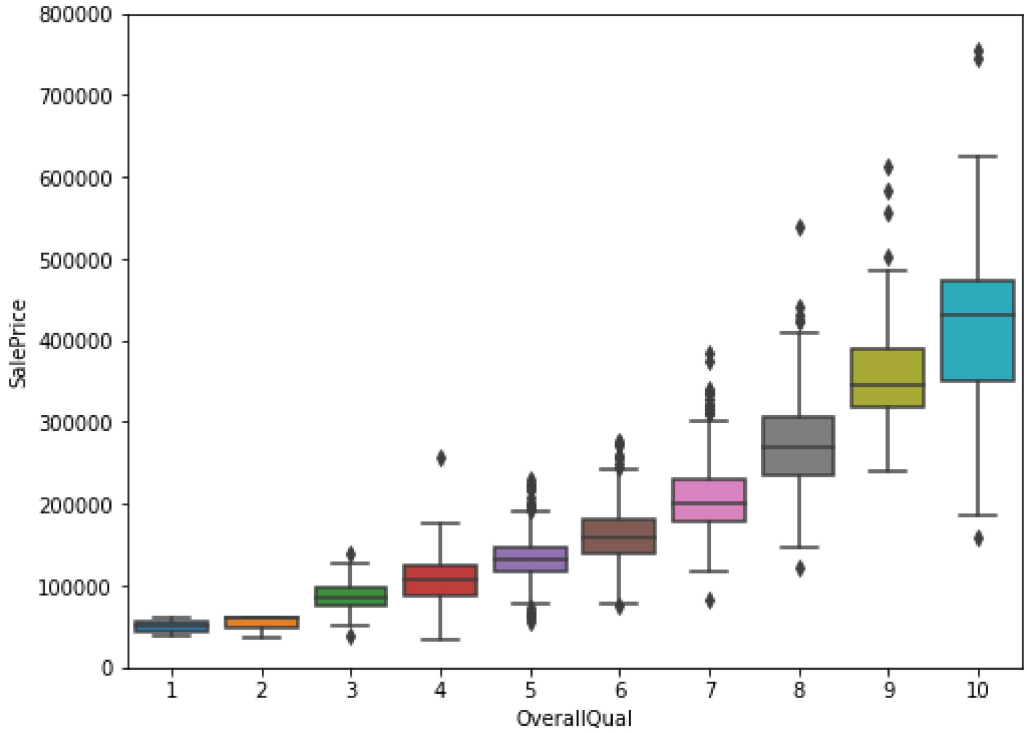


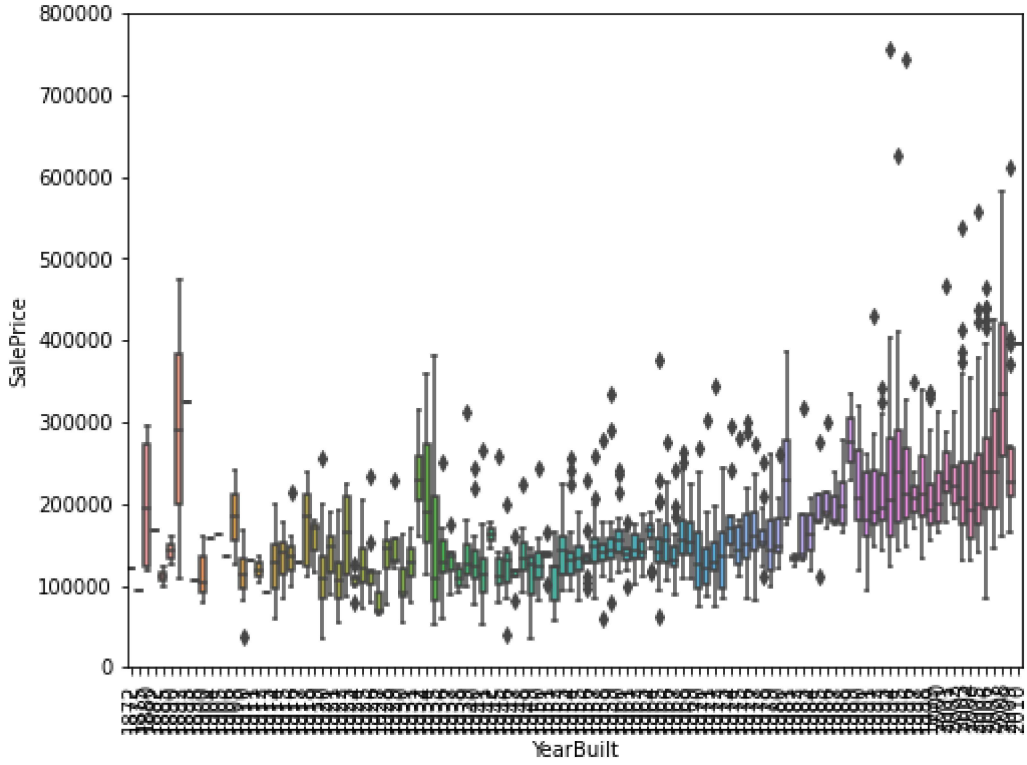

```
#Relationship with numerical variables by scatter plot TotalBsmtSF/saleprice
var = 'TotalBsmtSF'
data = pd.concat([df['SalePrice'], df[var]], axis=1)
data.plot.scatter(x=var, y='SalePrice', ylim=(0,800000));
```

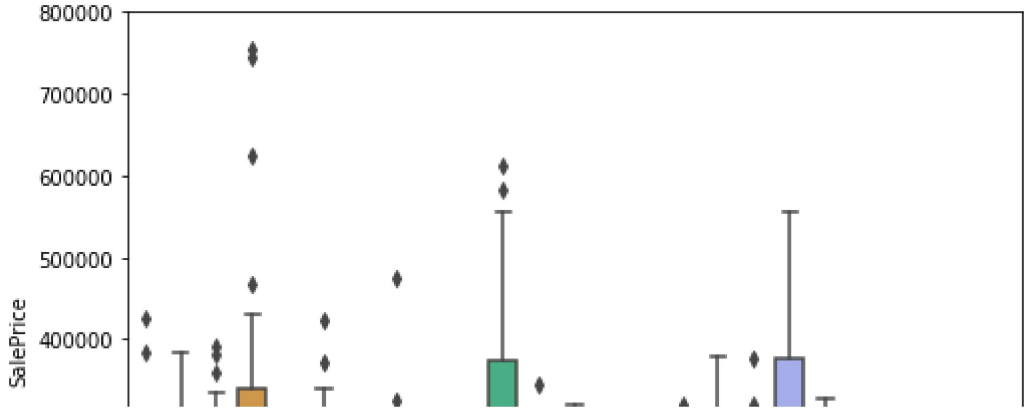

```
#Relationship with numerical variables by scatter plot GrLivArea/saleprice
var = 'GrLivArea'
data = pd.concat([df['SalePrice'], df[var]], axis=1)
data.plot.scatter(x=var, y='SalePrice', ylim=(0,800000));
```

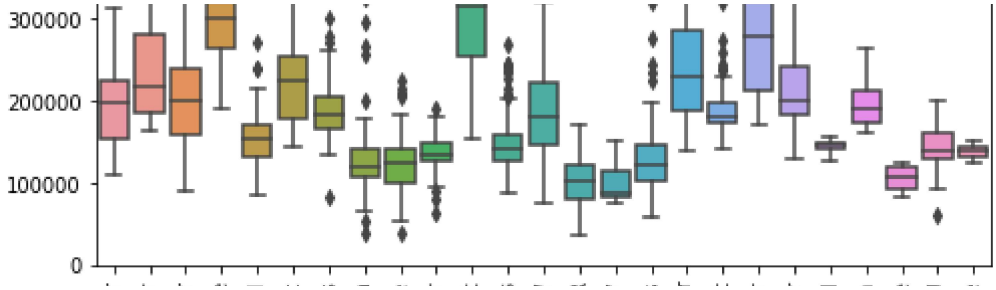



```
#Box Plot Relationship with categorical features using box plot overallqual/saleprice
var = 'OverallQual'
data = pd.concat([df['SalePrice'], df[var]], axis=1)
f, ax = plt.subplots(figsize=(8, 6))
fig = sns.boxplot(x=var, y="SalePrice", data=data)
fig.axis(ymin=0, ymax=800000);

#Relationship with categorical features using box plot YearBuilt/saleprice
var = 'YearBuilt'
data = pd.concat([df['SalePrice'], df[var]], axis=1)
f, ax = plt.subplots(figsize=(8, 6))
fig = sns.boxplot(x=var, y="SalePrice", data=data)
fig.axis(ymin=0, ymax=800000);
plt.xticks(rotation=90);

#Relationship with categorical features using box plot Neighborhood/saleprice
var = 'Neighborhood'
data = pd.concat([df['SalePrice'], df[var]], axis=1)
f, ax = plt.subplots(figsize=(8, 6))
fig = sns.boxplot(x=var, y="SalePrice", data=data)
fig.axis(ymin=0, ymax=800000);
plt.xticks(rotation=90);
```


800000
700000
600000
500000
400000
300000
200000
100000
0
SalePrice
1
2
3
4
5
6
7
8
9
10
OverallQual


800000
700000
600000
500000
400000
300000
200000
100000
0
SalePrice
YearBuilt


800000
700000
600000
500000
400000
SalePrice



GrLivArea' and 'TotalBsmtSF' seem to be linearly related with 'SalePrice' The box plot shows Sales prices increase with the Overall quality and throughout the Years Built. Sales prices is the highest in Neighbourhood NoRidge.

Task 5.Combine all the significant categorical and numerical variables

```
df.select_dtypes(include = 'number')
```

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	Year
0	1	60	65.0	8450	7	5	2003	
1	2	20	80.0	9600	6	8	1976	
2	3	60	68.0	11250	7	5	2001	
3	4	70	60.0	9550	7	5	1915	
4	5	60	84.0	14260	8	5	2000	
...	...	...	...	...	...	...	...	
1455	1456	60	62.0	7917	6	5	1999	
1456	1457	20	85.0	13175	6	6	1978	
1457	1458	70	66.0	9042	7	9	1941	
1458	1459	20	68.0	9717	5	6	1950	
1459	1460	20	75.0	9937	5	6	1965	

1460 rows × 38 columns

```
#Identify & Dropping variables with missing values

category_df_stats = pd.DataFrame(columns = ['column','num_miss', 'pct_miss'])

na_data = pd.DataFrame()

for c in category_df.columns:
    na_data['column'] = [c]
    na_data['num_miss'] = category_df[c].isnull().sum()
```

```
        na_data['pct_miss'] = (category_df[c].isnull().sum()/len(category_df)).round(3)*100
        category_df_stats = category_df_stats.append(na_data)

    category_df_stats
```

	column	num_miss	pct_miss
0	MSZoning	0	0.0
0	Street	0	0.0
0	Alley	1369	93.8
0	LotShape	0	0.0
0	LandContour	0	0.0
0	Utilities	0	0.0
0	LotConfig	0	0.0
0	LandSlope	0	0.0
0	Neighborhood	0	0.0
0	Condition1	0	0.0
0	Condition2	0	0.0
0	BldgType	0	0.0
0	HouseStyle	0	0.0
0	RoofStyle	0	0.0
0	RoofMatl	0	0.0

```
#Delete Summary Rows and Columns in the Dataset.
#Delete Extra Rows like blank rows, page numbers, etc.
#Delete List of variables with large number of missing values.

df_cols_rmv = ['Alley', 'FireplaceQu', 'PoolQC', 'Fence', 'MiscFeature']

# Remove the columns added to the df_cols_rmv list from df dataframe
df.drop(df_cols_rmv, axis = 1, inplace = True)

# reset index, to drop rows
df.reset_index(drop=True, inplace=True)

#Observe new data - There's no more missing values
df.head(2)
```

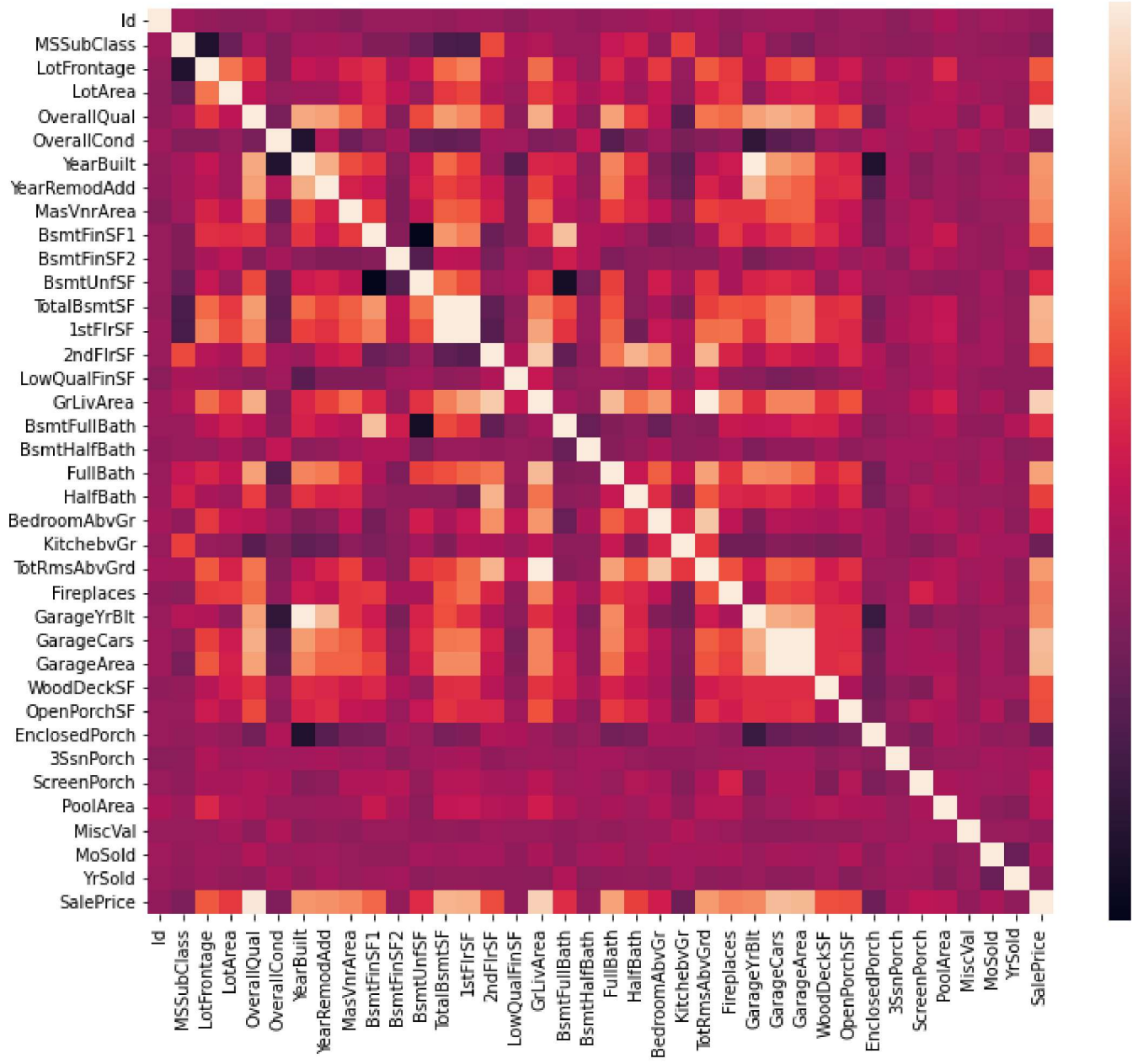
	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	LotShape	LandContour	Utilit
0	1	60	RL	65.0	8450	Pave	Reg	Lvl	All
1	2	20	RL	80.0	9600	Pave	Reg	Lvl	All

2 rows × 76 columns

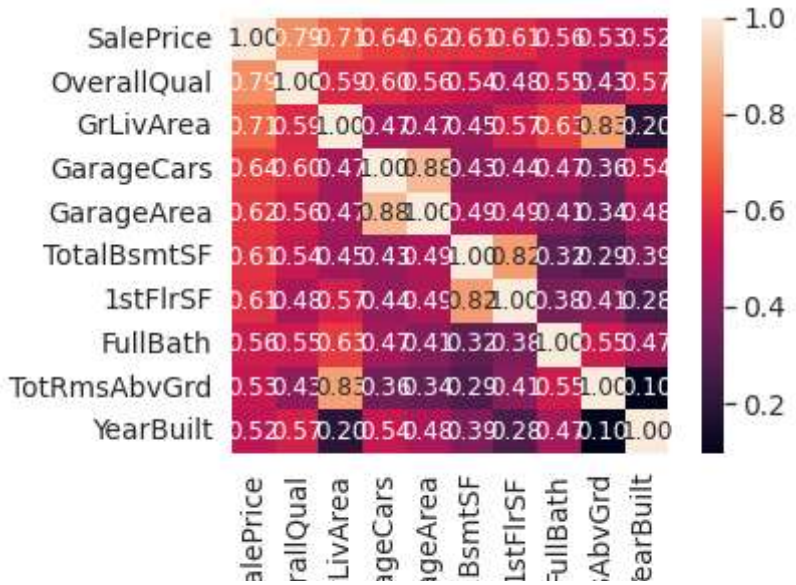
```
#get df shape
print('\nDataFrame Shape: ',category_df.shape)
print('----------------------------------------')
df.select_dtypes(include = 'object').info()
```

```
DataFrame Shape:  (1460, 43)
----------------------------------------
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 38 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   MSZoning       1460 non-null   object
 1   Street         1460 non-null   object
 2   LotShape       1460 non-null   object
 3   LandContour    1460 non-null   object
 4   Utilities      1460 non-null   object
 5   LotConfig      1460 non-null   object
 6   LandSlope      1460 non-null   object
 7   Neighborhood   1460 non-null   object
 8   Condition1     1460 non-null   object
 9   Condition2     1460 non-null   object
 10  BldgType       1460 non-null   object
 11  HouseStyle     1460 non-null   object
 12  RoofStyle      1460 non-null   object
 13  RoofMatl       1460 non-null   object
 14  Exterior1st    1460 non-null   object
 15  Exterior2nd    1460 non-null   object
 16  MasVnrType     1452 non-null   object
 17  ExterQual      1460 non-null   object
 18  ExterCond      1460 non-null   object
 19  Foundation     1460 non-null   object
 20  BsmtQual       1423 non-null   object
 21  BsmtCond       1423 non-null   object
 22  BsmtExposure   1422 non-null   object
 23  BsmtFinType1   1423 non-null   object
 24  BsmtFinType2   1422 non-null   object
 25  Heating        1460 non-null   object
 26  HeatingQC      1460 non-null   object
 27  CentralAir     1460 non-null   object
 28  Electrical     1459 non-null   object
 29  KitchenQual    1460 non-null   object
 30  Functiol       1460 non-null   object
 31  GarageType     1379 non-null   object
 32  GarageFinish   1379 non-null   object
 33  GarageQual     1379 non-null   object
 34  GarageCond     1379 non-null   object
 35  PavedDrive     1460 non-null   object
 36  SaleType       1460 non-null   object
 37  SaleCondition  1460 non-null   object
dtypes: object(38)
memory usage: 433.6+ KB
```

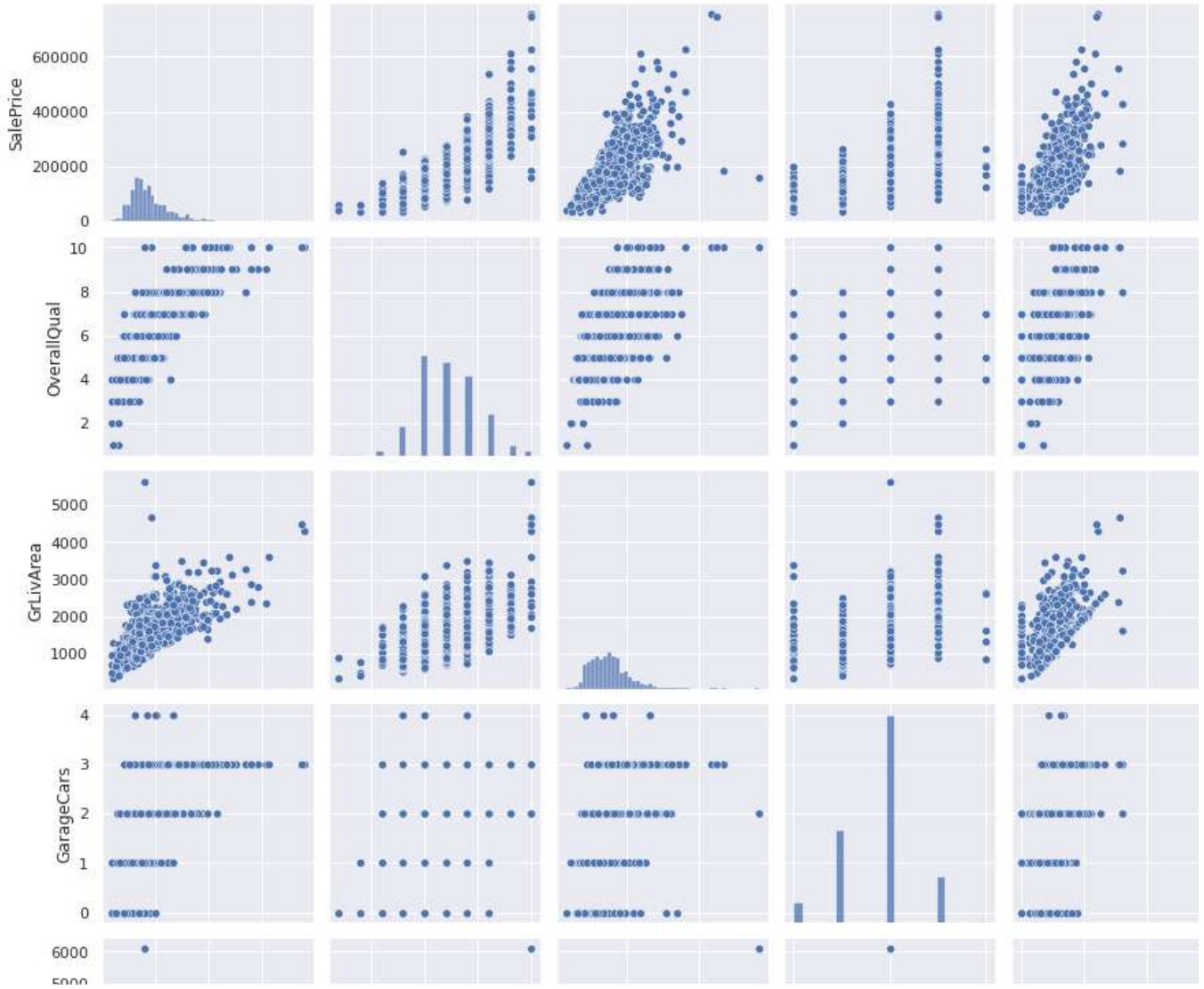
```
#correlation matrix
corrmat = df.corr()
f, ax = plt.subplots(figsize=(12, 10))
sns.heatmap(corrmat, vmax=.8, square=True);
```



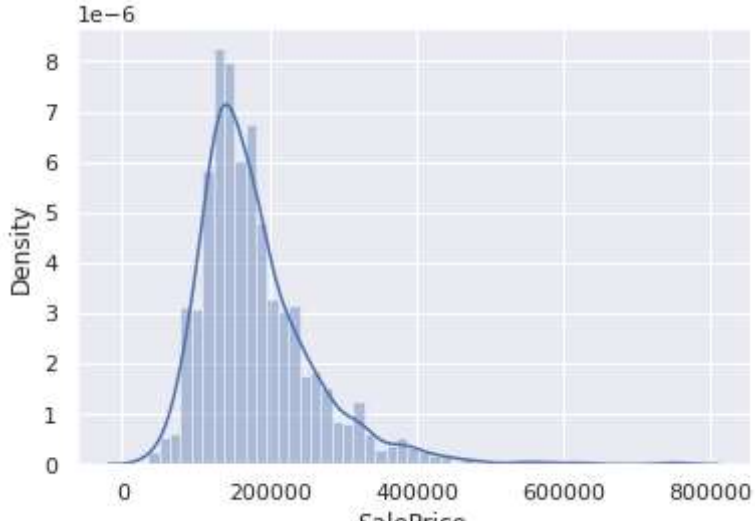

```
#saleprice correlation matrix
k = 10 #number of variables for heatmap
cols = corrmat.nlargest(k, 'SalePrice')['SalePrice'].index
cm = np.corrcoef(df[cols].values.T)
sns.set(font_scale=1.25)
hm = sns.heatmap(cm, cbar=True, annot=True, square=True, fmt='.2f', annot_kws={'size': 12}, y
plt.show()
```




```
#Scatter plots between 'SalePrice' and correlated variables
sns.set()
cols = ['SalePrice', 'OverallQual', 'GrLivArea', 'GarageCars', 'TotalBsmtSF', 'FullBath', 'Ye
sns.pairplot(df[cols], size = 2.5)
plt.show();
```




```
#histogram and normal probability plot
sns.distplot(df['SalePrice']);
fig = plt.figure()
res = stats.probplot(df['SalePrice'], plot=plt)
```




```
sns.countplot(df['SalePrice'], color="salmon", facecolor=(0, 0, 0, 0), linewidth=5, edgecolor
plt.show()
```

Task 6. Plot box plot for the new dataset to find the variables with outliers

```
category_df.head(2)
```

	MSZoning	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Nei
0	RL	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	
1	RL	Pave	NaN	Reg	Lvl	AllPub	FR2	Gtl	

2 rows × 43 columns

```
numeric_df.head(2)
```

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodA
0	1	60	65.0	8450	7	5	2003	20
1	2	20	80.0	9600	6	8	1976	19

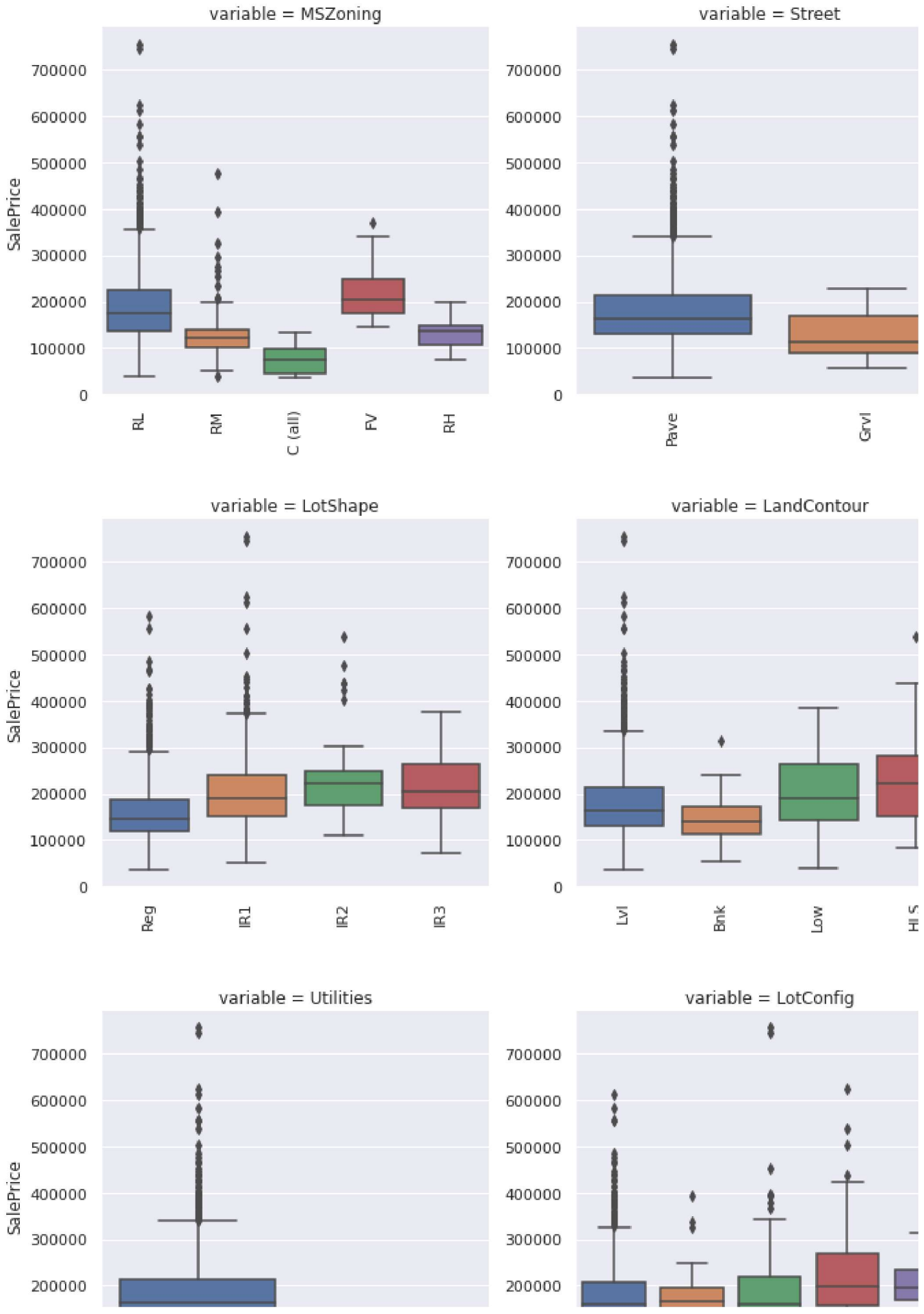
2 rows × 38 columns

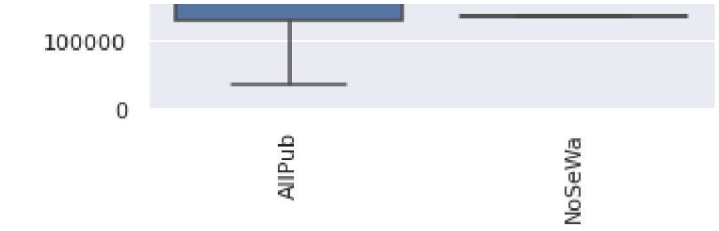
```
norm_data = np.random.normal(size=100000)
skewed_data = np.concatenate((np.random.normal(size=35000)+2,
                              np.random.exponential(size=65000)),
                              axis=0)
uniform_data = np.random.uniform(0,2, size=100000)
peaked_data = np.concatenate((np.random.exponential(size=50000),
                              np.random.exponential(size=50000)*(-1)),
                              axis=0)

data_df = pd.DataFrame({"norm":norm_data,
                        "skewed":skewed_data,
                        "uniform":uniform_data,
                        "peaked":peaked_data})
data_df.plot(kind="density",
             figsize=(10,10),
             xlim=(-5,5))
```

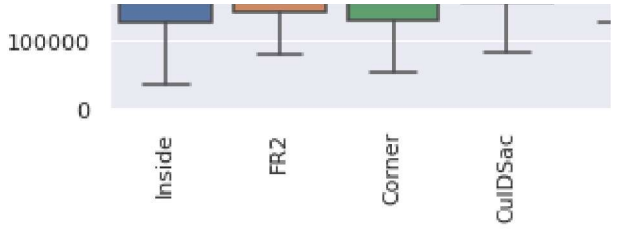
```
<AxesSubplot:ylabel='Density'>
```

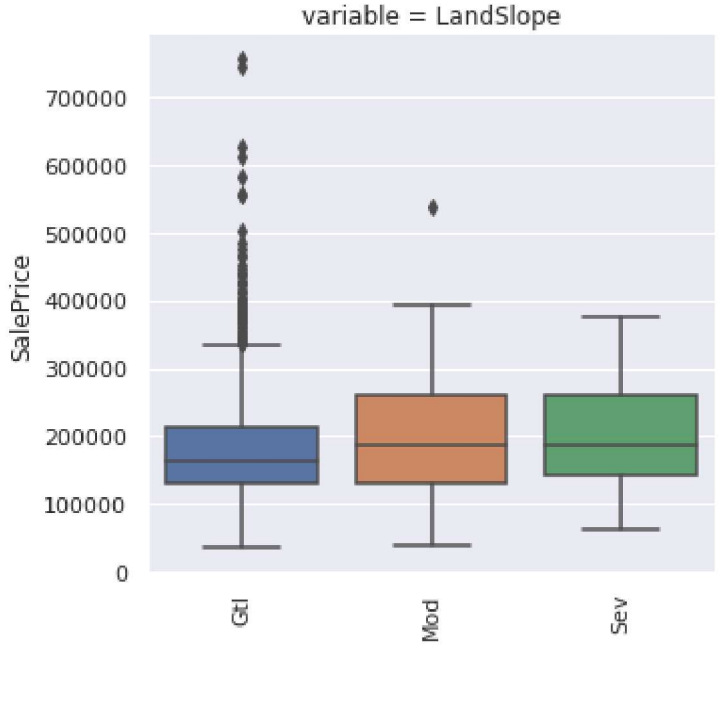

```
#Create boxplots for visualizing categorical variables.
def boxplot(x,y,**kwargs):
            sns.boxplot(x=x,y=y)
            x = plt.xticks(rotation=90)

cat = [f for f in df.columns if df.dtypes[f] == 'object']

p = pd.melt(df, id_vars='SalePrice', value_vars=cat)
g = sns.FacetGrid (p, col='variable', col_wrap=2, sharex=False, sharey=False, size=5)
g = g.map(boxplot, 'value','SalePrice')
g
```

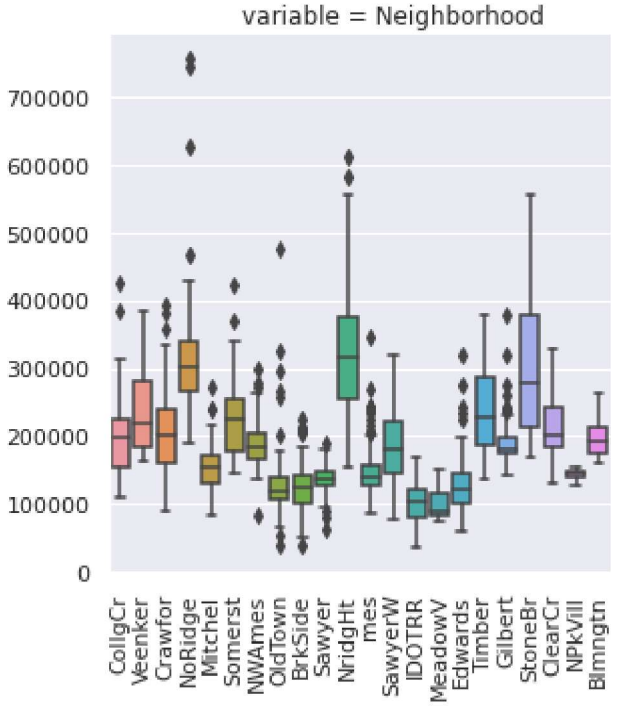
```
<seaborn.axisgrid.FacetGrid at 0x7f3ba6cc8c90>
```

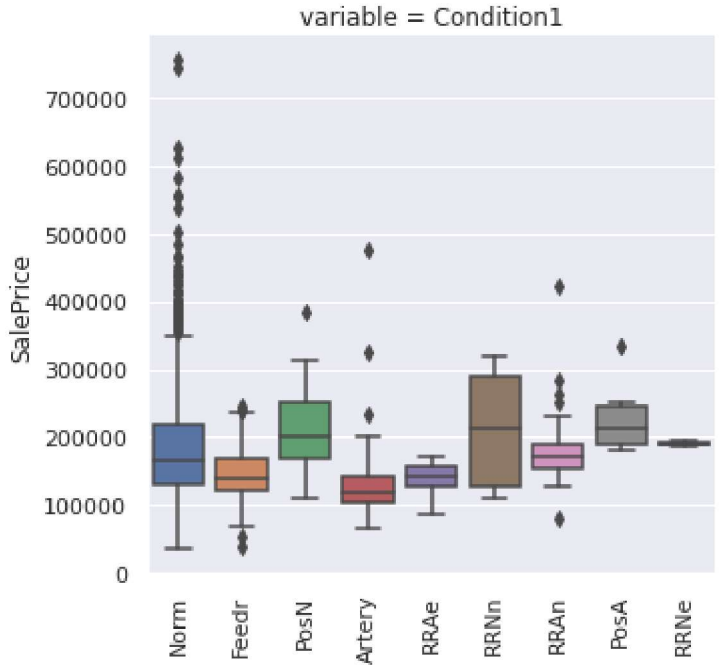


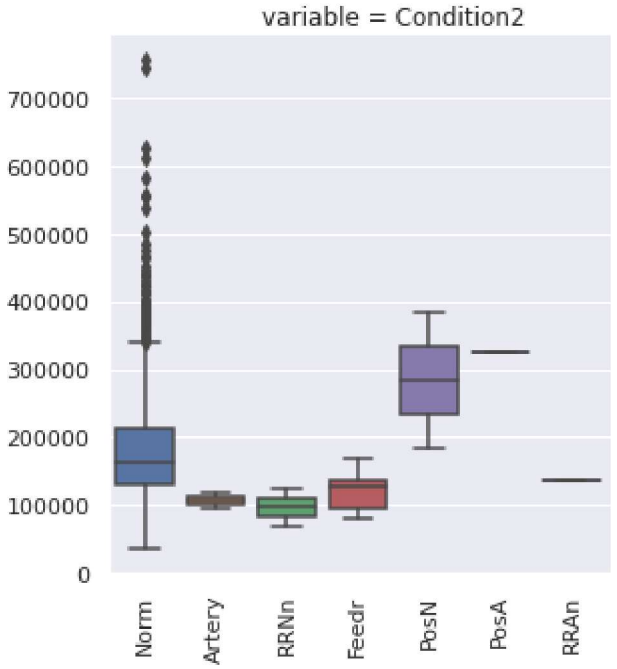

100000
0
AllPub
NoSeWa

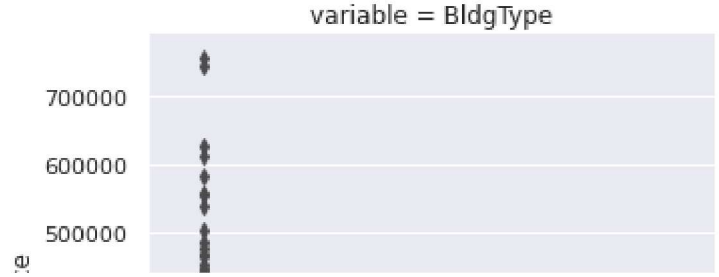

100000
0
Inside
FR2
Corner
CulDSac

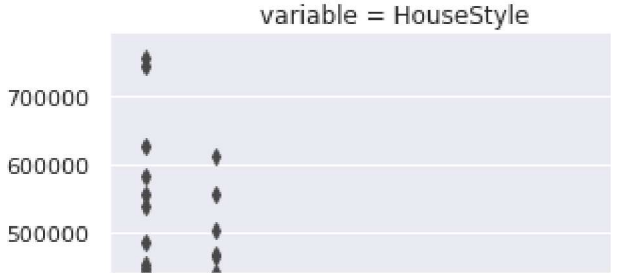

variable = LandSlope
SalePrice
700000
600000
500000
400000
300000
200000
100000
0
Gtl
Mod
Sev

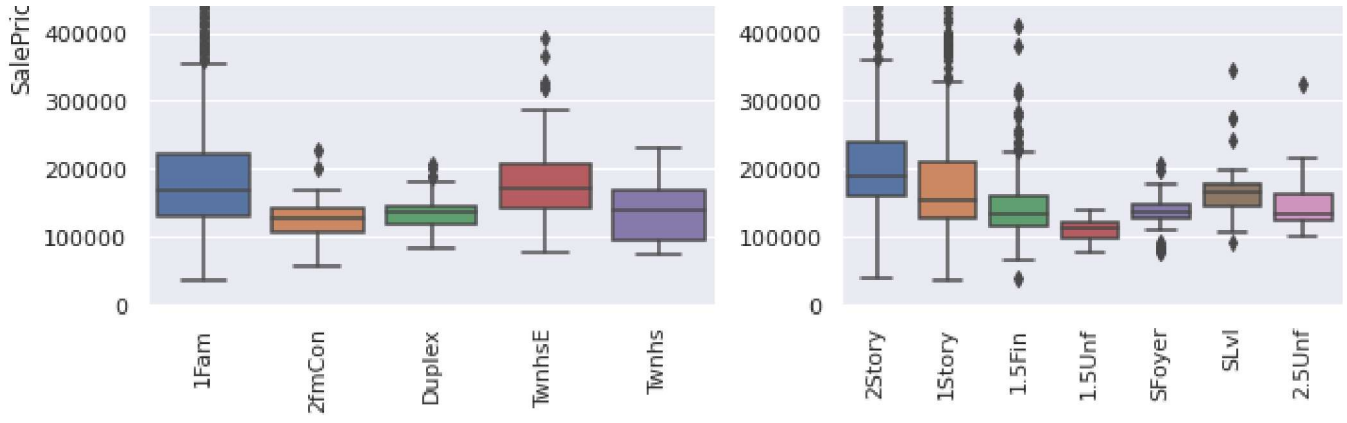

variable = Neighborhood
700000
600000
500000
400000
300000
200000
100000
0
CollgCr
Veenker
Crawfor
NoRidge
Mitchel
Somerst
NWAmes
OldTown
BrkSide
Sawyer
NridgHt
mes
SawyerW
IDOTRR
MeadowV
Edwards
Timber
Gilbert
StoneBr
ClearCr
NPkVill
Blmngtn


variable = Condition1
SalePrice
700000
600000
500000
400000
300000
200000
100000
0
Norm
Feedr
PosN
Artery
RRAe
RRNn
RRAn
PosA
RRNe


variable = Condition2
700000
600000
500000
400000
300000
200000
100000
0
Norm
Artery
RRNn
Feedr
PosN
PosA
RRAn

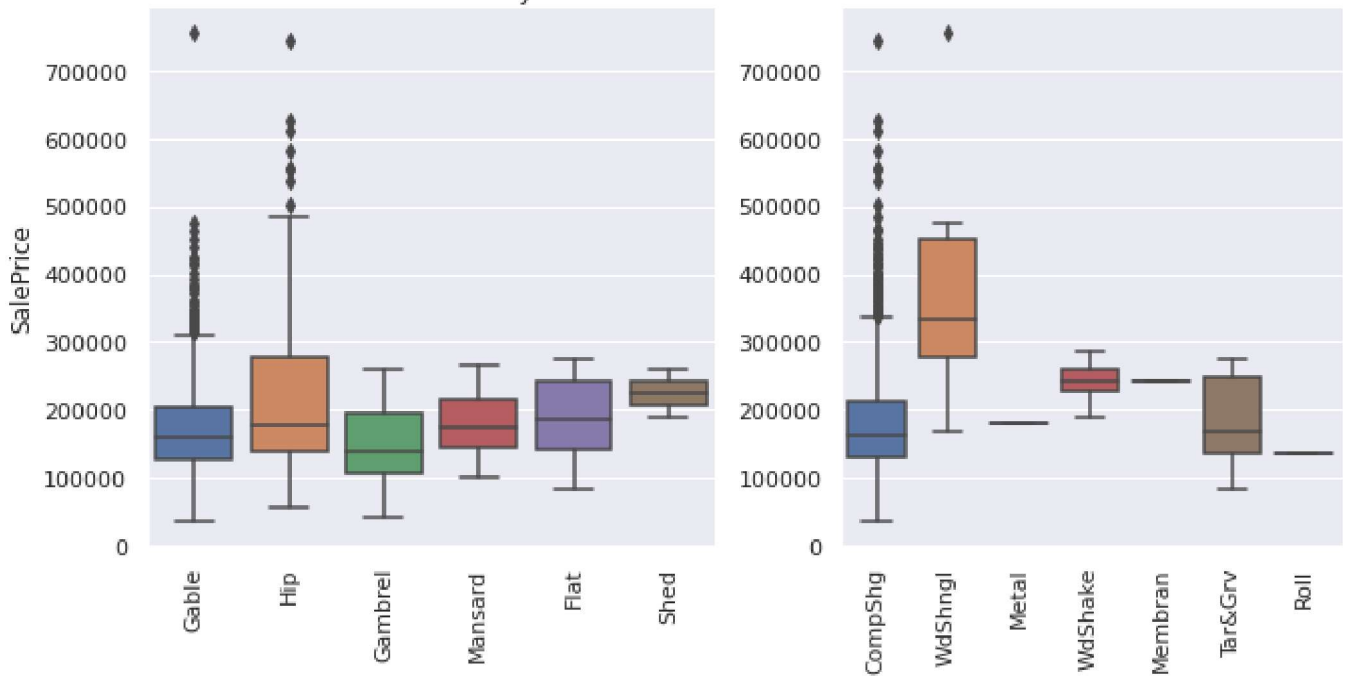

variable = BldgType
700000
600000
500000


variable = HouseStyle
700000
600000
500000



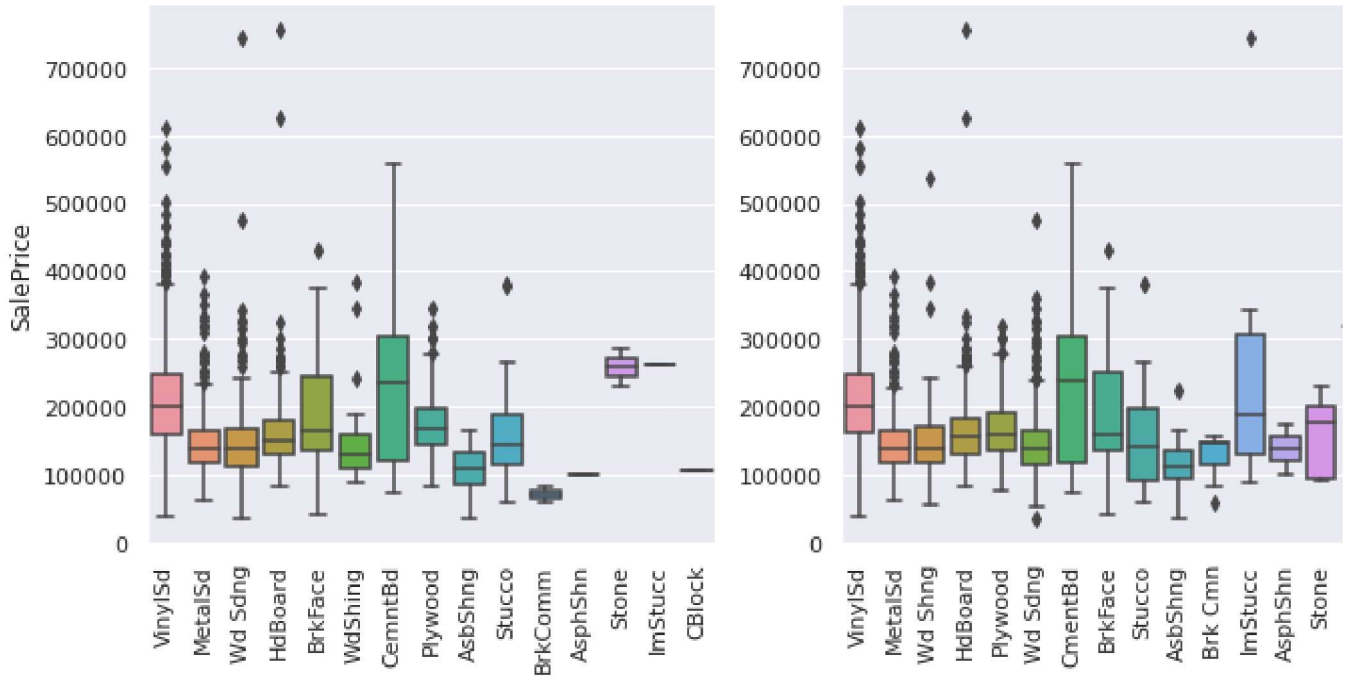
variable = RoofStyle

variable = RoofMatl



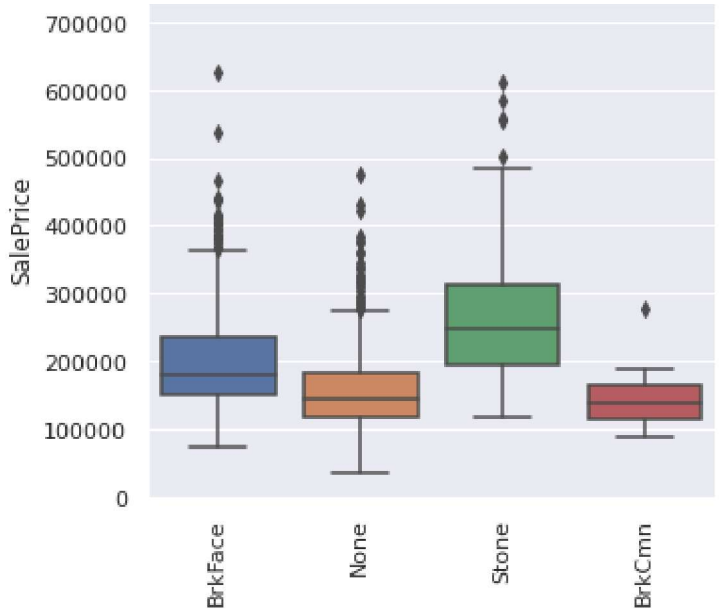
variable = Exterior1st

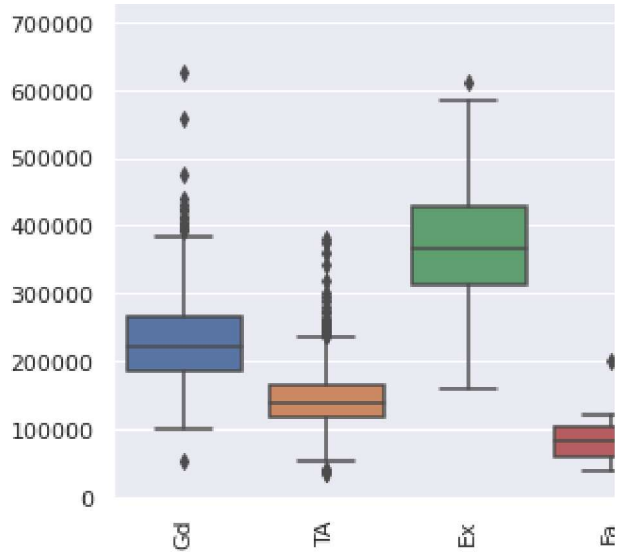
variable = Exterior2nd

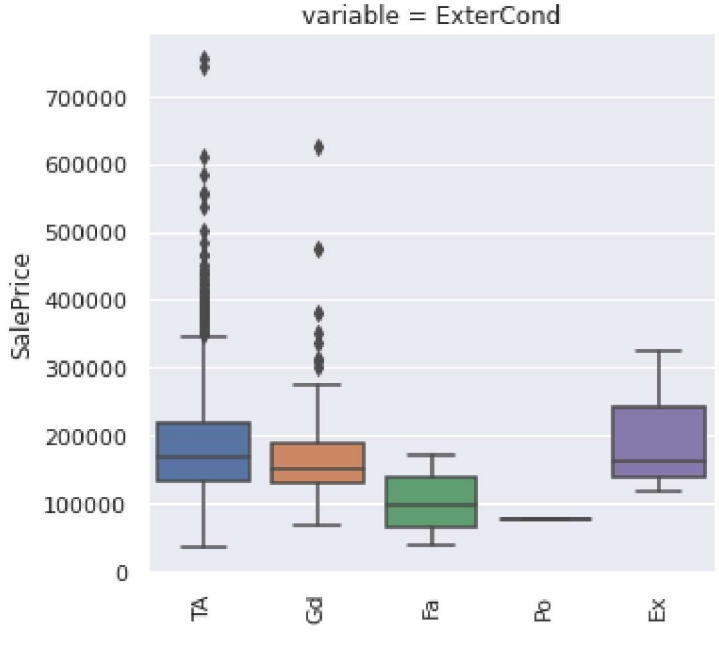


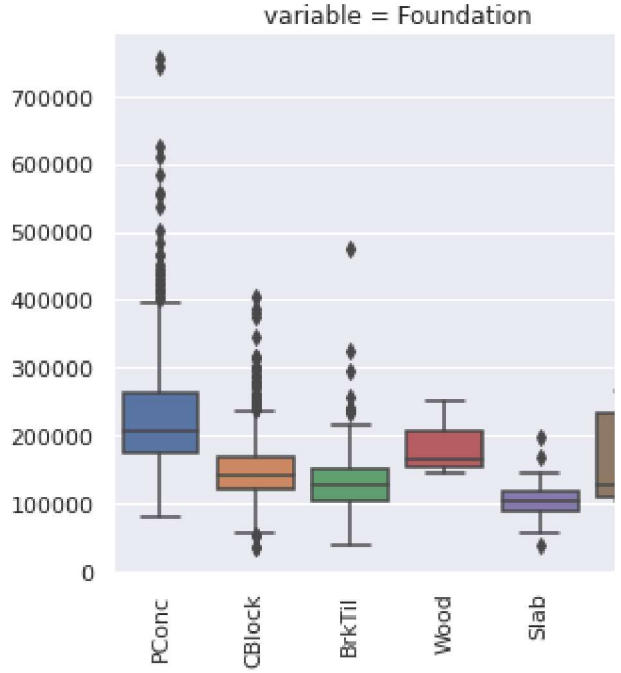
variable = MasVnrType

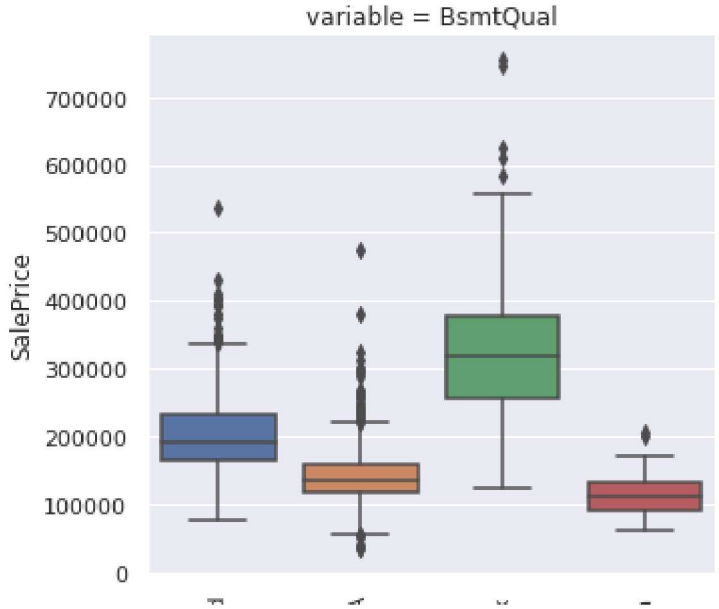
variable = ExterQual

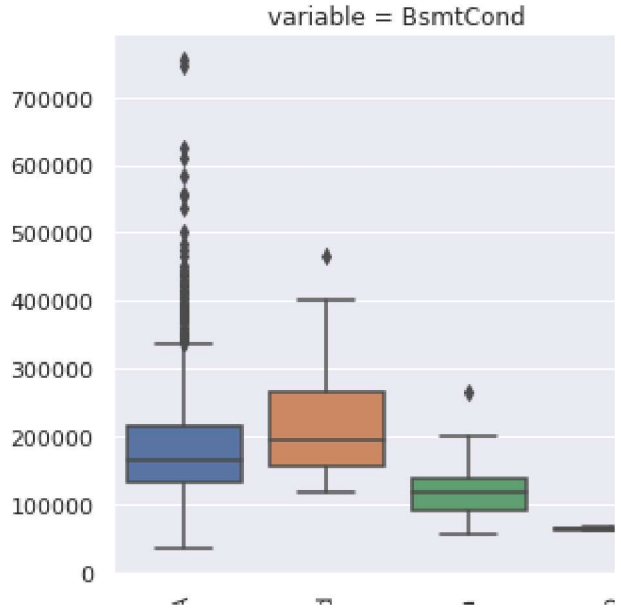

SalePrice
700000
600000
500000
400000
300000
200000
100000
0
BrkFace
None
Stone
BrkCmn


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


variable = ExterCond
SalePrice
700000
600000
500000
400000
300000
200000
100000
0
TA
Gd
Fa
Po
Ex


variable = Foundation
700000
600000
500000
400000
300000
200000
100000
0
PConc
CBlock
BrkTil
Wood
Slab


variable = BsmtQual
SalePrice
700000
600000
500000
400000
300000
200000
100000
0


variable = BsmtCond
700000
600000
500000
400000
300000
200000
100000
0