eda

May 19, 2021

1 3. Exploratory Data Analysis

1.1 3.1 Read data

[29]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import norm
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer
from scipy import stats
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

[30]:
```
df = pd.read_csv('train.csv')
```

[3]:
```
df.head()
```

[3]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	\
0	1	60	RL	65.0	8450	Pave	NaN	Reg	
1	2	20	RL	80.0	9600	Pave	NaN	Reg	
2	3	60	RL	68.0	11250	Pave	NaN	IR1	
3	4	70	RL	60.0	9550	Pave	NaN	IR1	
4	5	60	RL	84.0	14260	Pave	NaN	IR1	

	LandContour	Utilities	…	PoolArea	PoolQC	Fence	MiscFeature	MiscVal	MoSold	\
0	Lvl	AllPub	…	0	NaN	NaN	NaN	0	2	
1	Lvl	AllPub	…	0	NaN	NaN	NaN	0	5	
2	Lvl	AllPub	…	0	NaN	NaN	NaN	0	9	
3	Lvl	AllPub	…	0	NaN	NaN	NaN	0	2	
4	Lvl	AllPub	…	0	NaN	NaN	NaN	0	12	

	YrSold	SaleType	SaleCondition	SalePrice
0	2008	WD	Normal	208500
1	2007	WD	Normal	181500
2	2008	WD	Normal	223500

```
3   2006        WD         Abnorml      140000
4   2008        WD          Normal      250000

[5 rows x 81 columns]
```

[4]:
```
df.groupby('SaleCondition')['LotFrontage','YrSold'].mean()
```

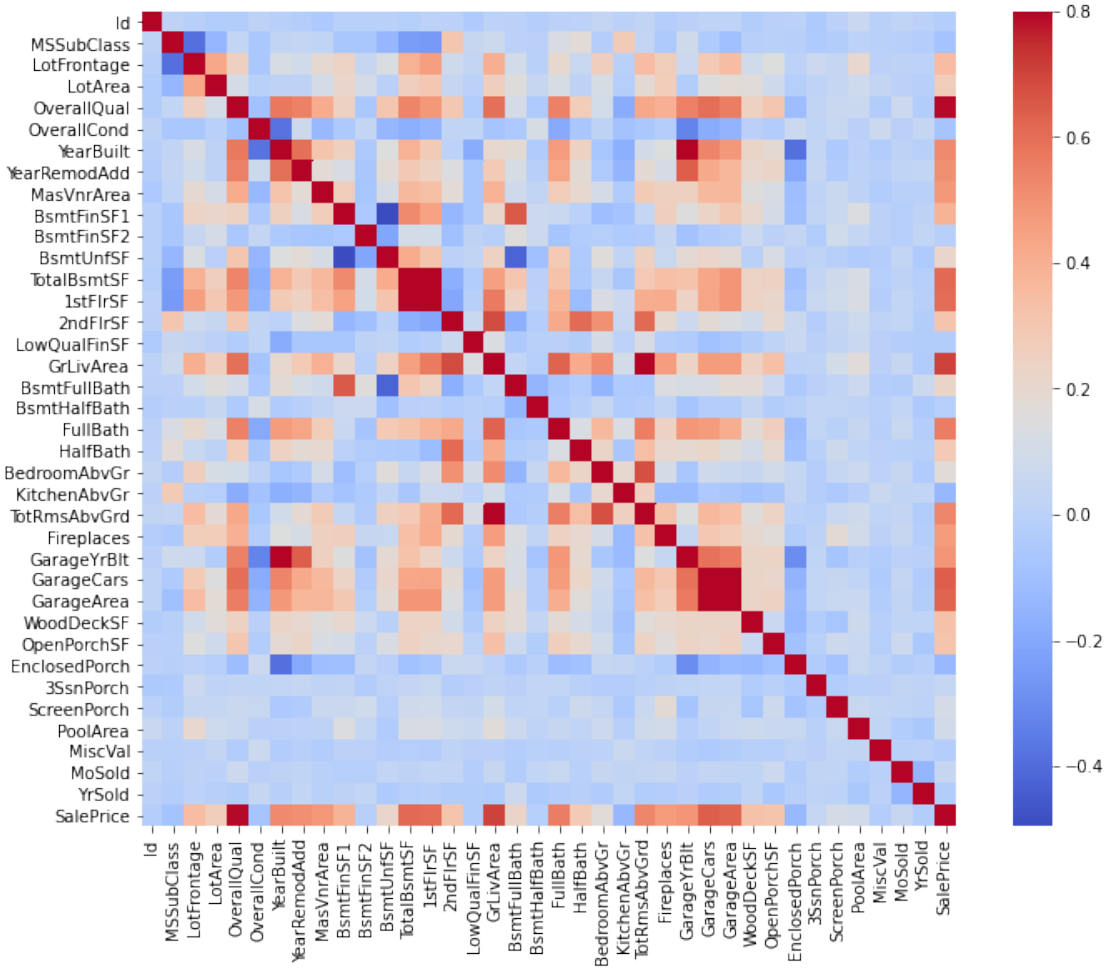
[4]:
```
                LotFrontage       YrSold
SaleCondition
Abnorml           67.928571  2007.663366
AdjLand           54.500000  2006.750000
Alloca            64.800000  2008.166667
Family            73.333333  2007.200000
Normal            69.124870  2007.897329
Partial           79.104839  2007.256000
```

1.2 3.2 Correlation Matrix

1.2.1 3.2.1 Correlation Matrix with heatmap

[5]:
```
# Correlation matrix
corrmat = df.corr()
f, ax = plt.subplots(figsize=(12, 9))
sns.heatmap(corrmat, vmax=.8, square=True,cmap="coolwarm")
```

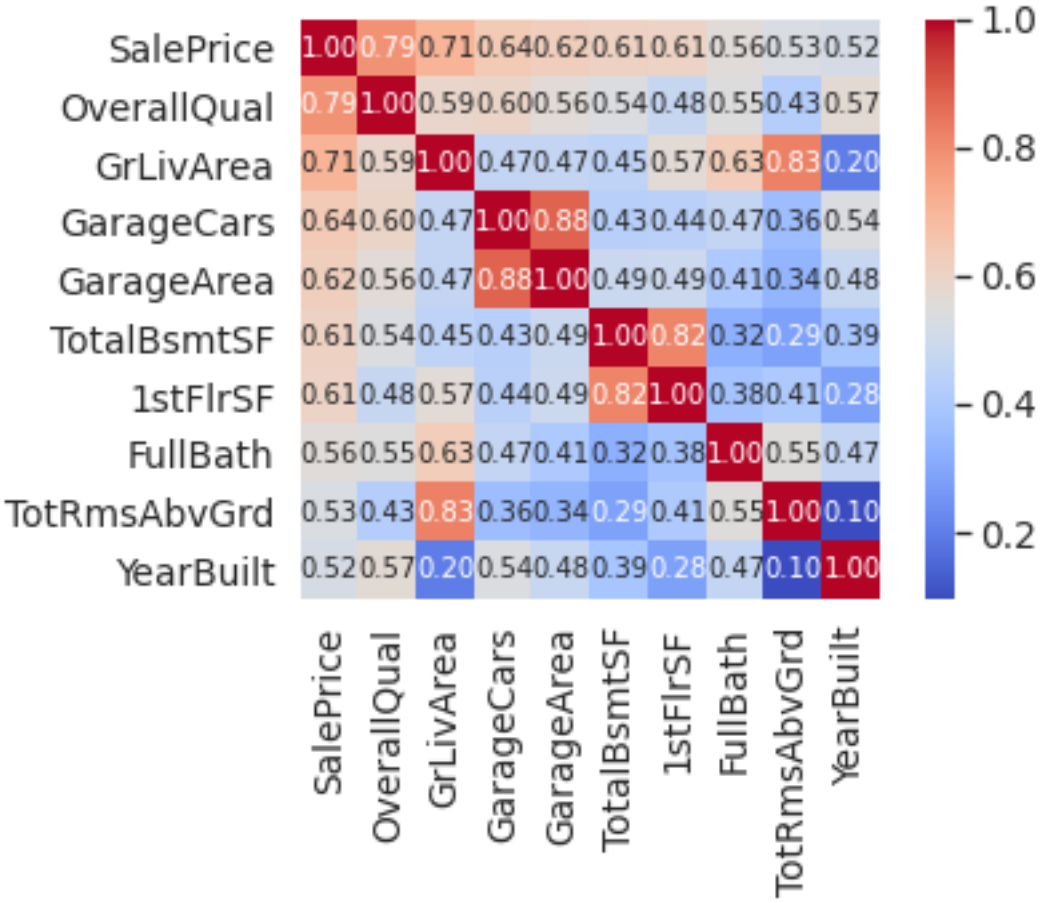
[5]:
```
<AxesSubplot:>
```



From above matrix we can see that features like OverallQual, GrLivArea, GarageArea, GarageCars,YearBuilt etc have strong correlation with SalesPrice.

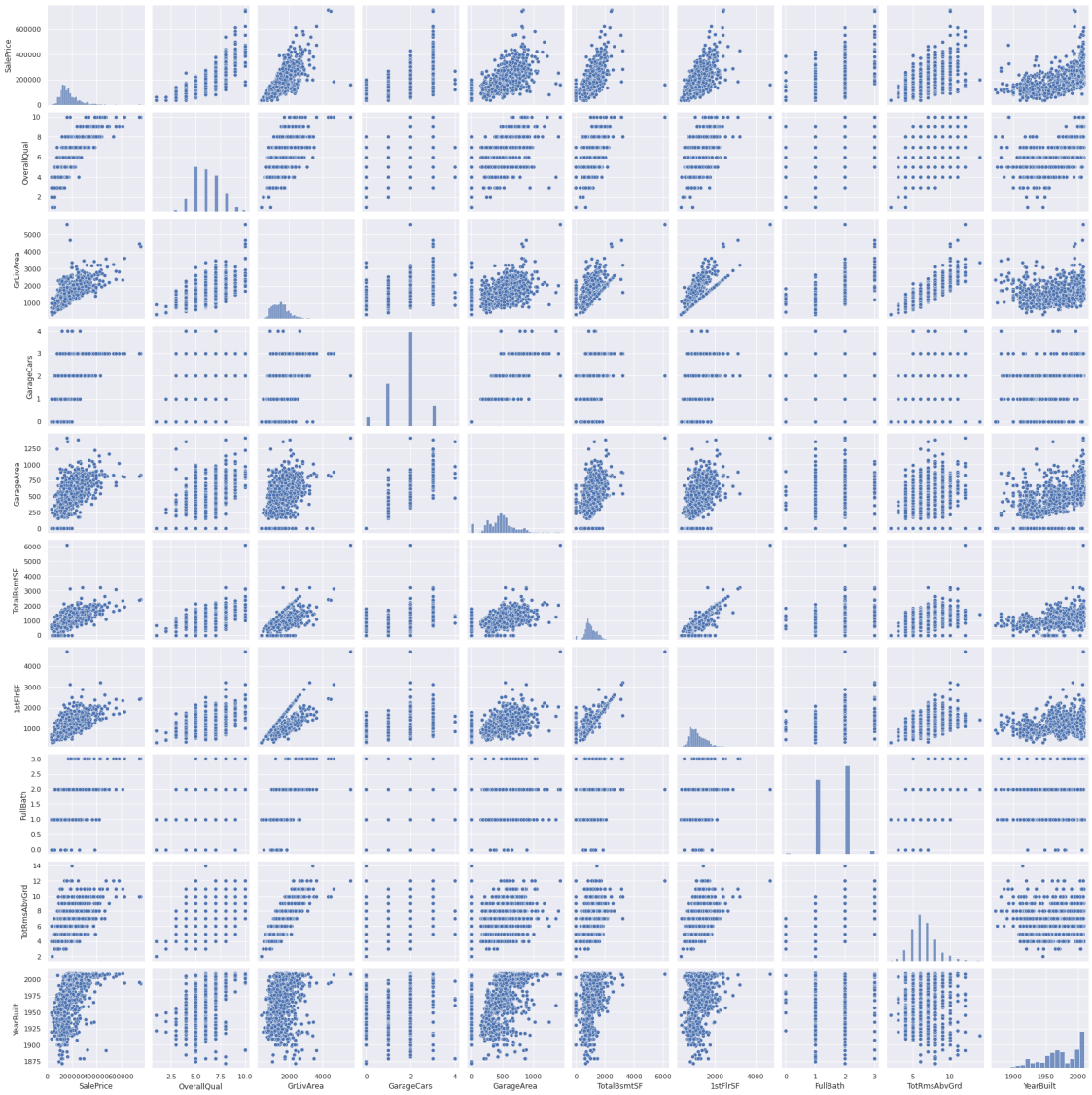
1.2.2 3.2.2 Top correlations with sales price

[6]:
```
# Top feature correlation with SalesPrice
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': 10}, yticklabels=cols.values, xticklabels=cols.values,␣
 ↪cmap="coolwarm")
plt.show()
```



1.3 3.3 Pair Plots

[7]:
```
#scatterplot
sns.set()
# cols = ['SalePrice', 'OverallQual', 'GrLivArea', 'GarageArea','GarageCars',␣
 ↪'TotalBsmtSF', 'FullBath', 'YearBuilt']
sns.pairplot(df[cols], height = 2.5)
plt.show();
```

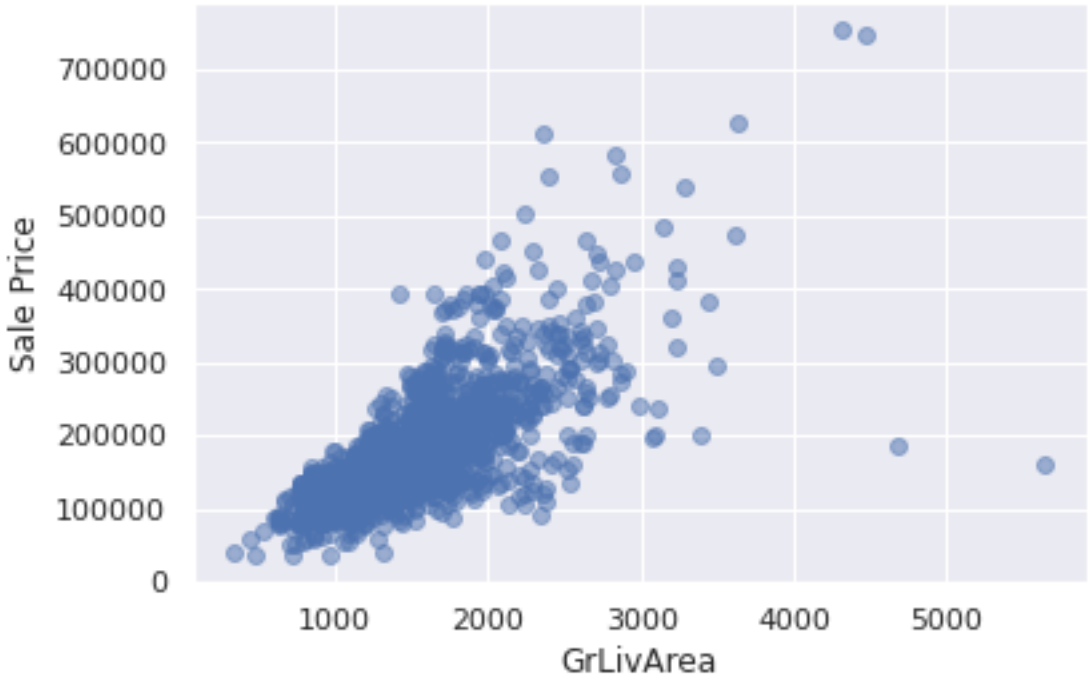


1.4 3.4 Scatter Plots

[8]:
```
# functuion to plot scatter plot
def scatterPlot(feature):
    plt.scatter(df[feature], df['SalePrice'], alpha=0.5)
    plt.xlabel(feature)
    plt.ylabel("Sale Price")
    plt.show()
```

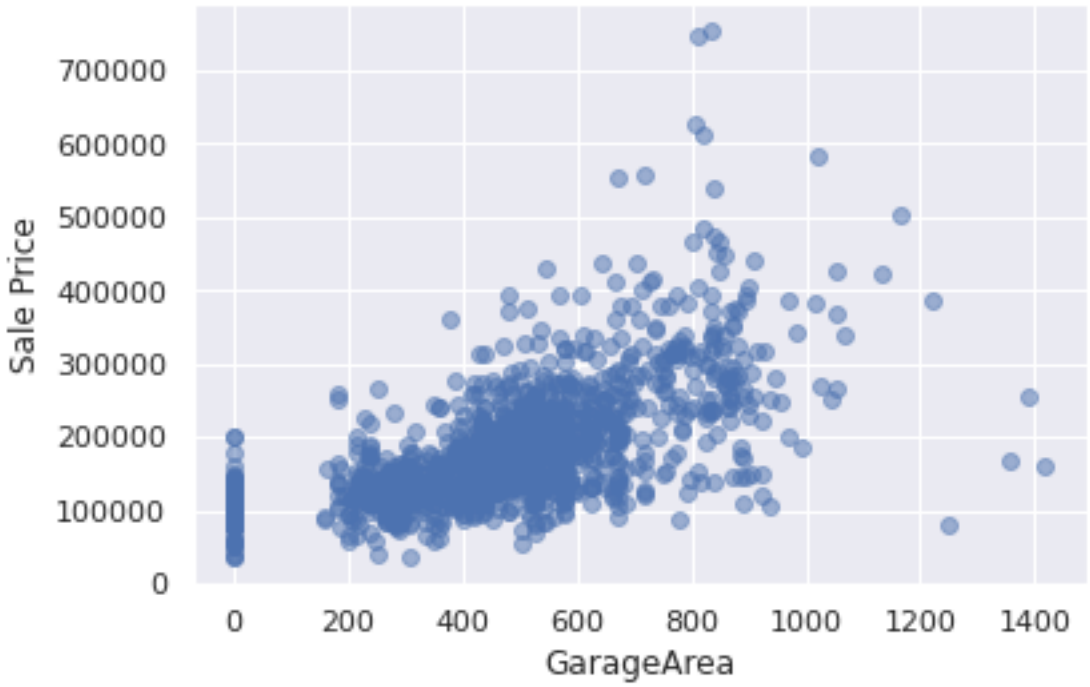
1.4.1 3.4.1 Ground living area

[9]:
```
scatterPlot('GrLivArea')
```



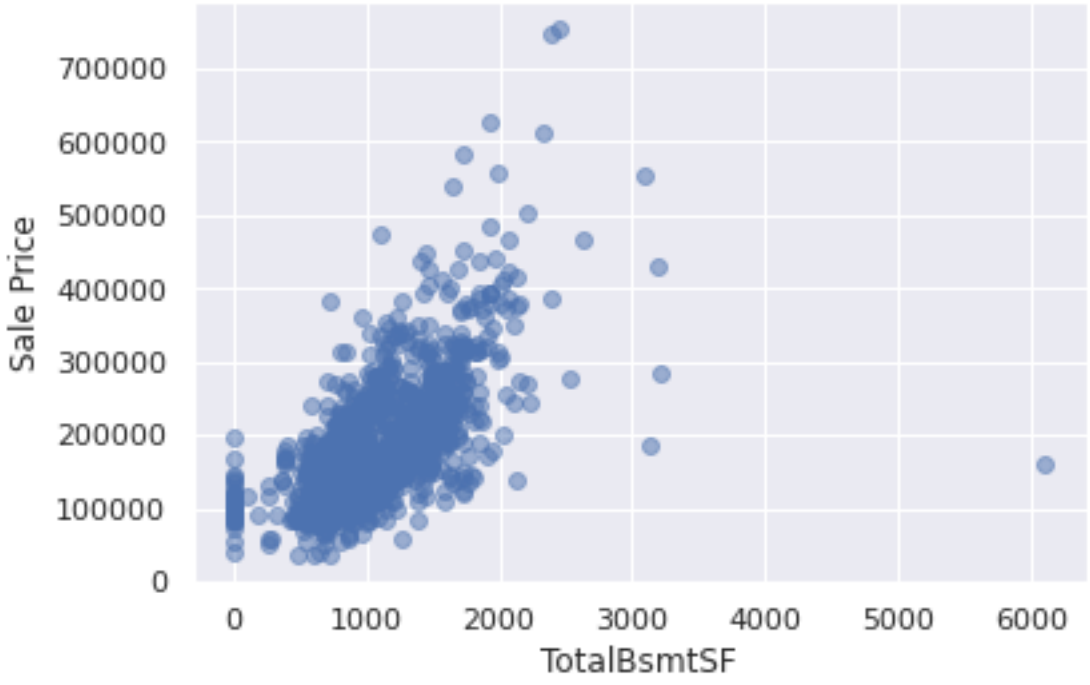
1.4.2 3.4.2 Garage Area

[10]:
```
scatterPlot('GarageArea')
```



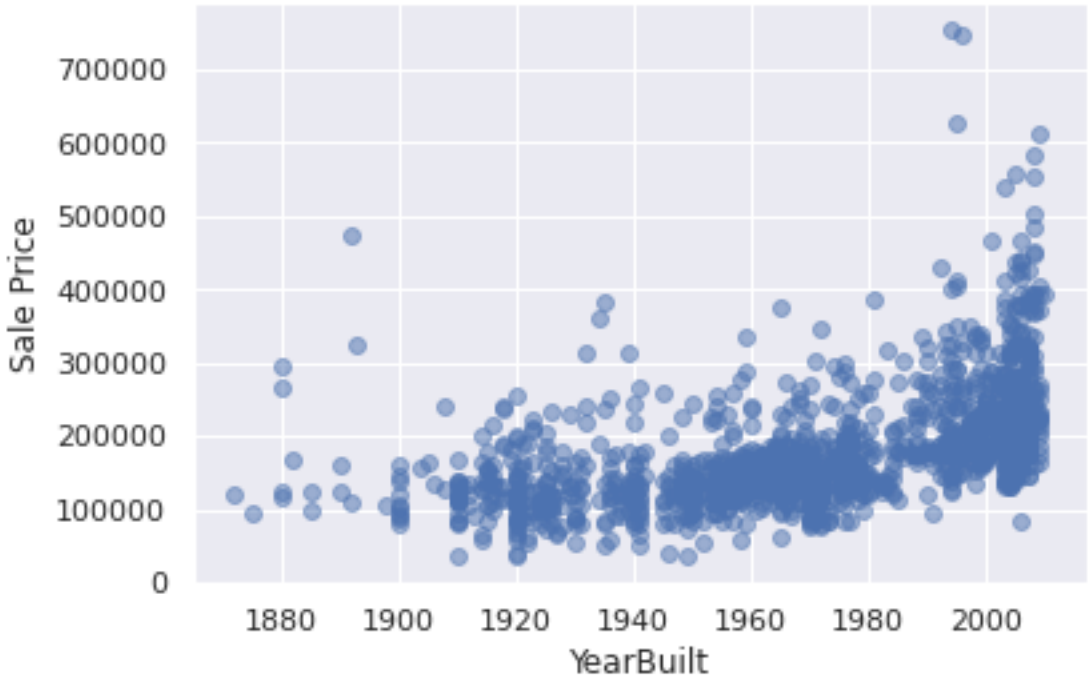
1.4.3 3.4.3 First Floor square feet

[11]:
```
scatterPlot('TotalBsmtSF')
```



1.4.4 3.4.5 Year built

[12]:
```
scatterPlot('YearBuilt')
```



From above plots we can see each of these features have a somewhat linear relationship with Sales Price.

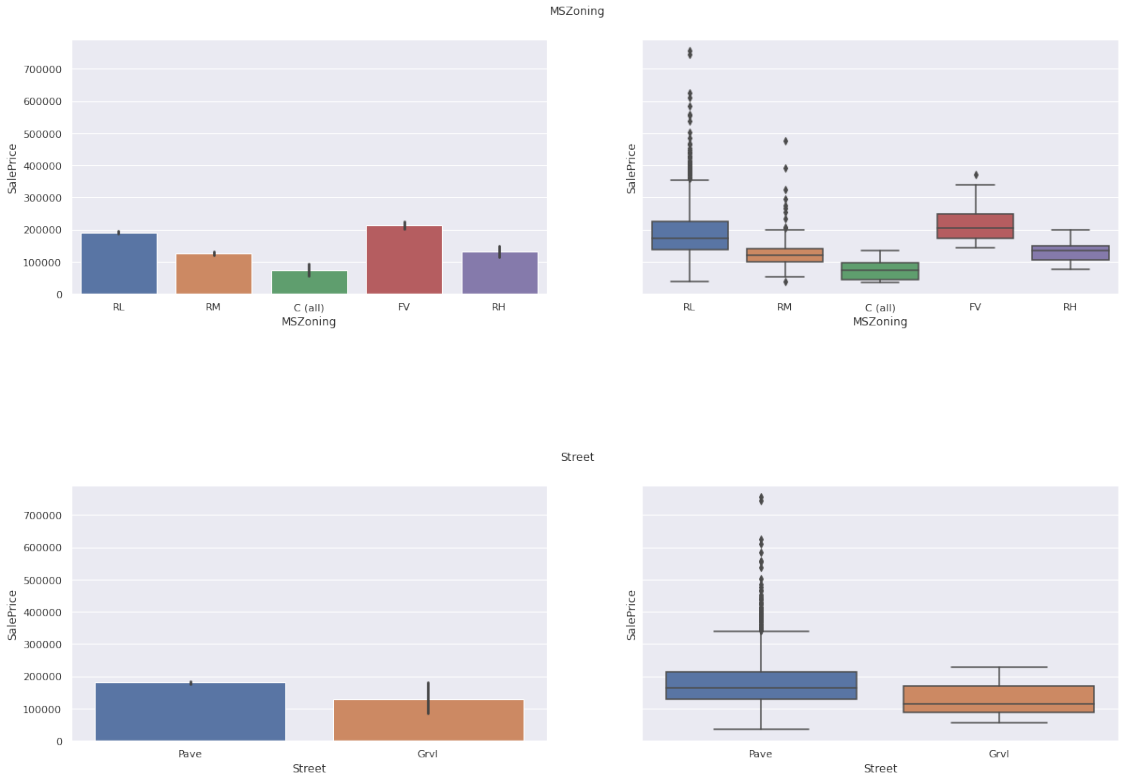
1.5 3.5 Features analysis

[26]:
```
nominal = [ 'MSZoning' ,'Street' ,'Alley' ,'LotShape' ,'Utilities' ,'LotConfig'␣
 ↪,'Neighborhood' ,'RoofStyle' ,'RoofMatl' ,'Exterior1st' ,'Exterior2nd'␣
 ↪,'MasVnrType' ,'Foundation' ,'Heating' ,'CentralAir' ,'Electrical'␣
 ↪,'GarageType'  ,'SaleType' ,'SaleCondition','MiscFeature']

ordinal = ['LandContour', 'LandSlope' ,'Condition1' ,'Condition2' ,'BldgType'␣
 ↪,'HouseStyle' ,'OverallQual' ,'OverallCond' ,'ExterQual' ,'ExterCond'␣
 ↪,'BsmtQual' ,'BsmtCond' ,'BsmtExposure' ,'BsmtFinType1' ,'HeatingQC'␣
 ↪,'FireplaceQu' ,'GarageFinish' ,'GarageCars' ,'GarageQual' ,'GarageCond'␣
 ↪,'PoolQC' ,'Fence', 'BsmtFinType2', 'KitchenQual','Functional','PavedDrive']
```

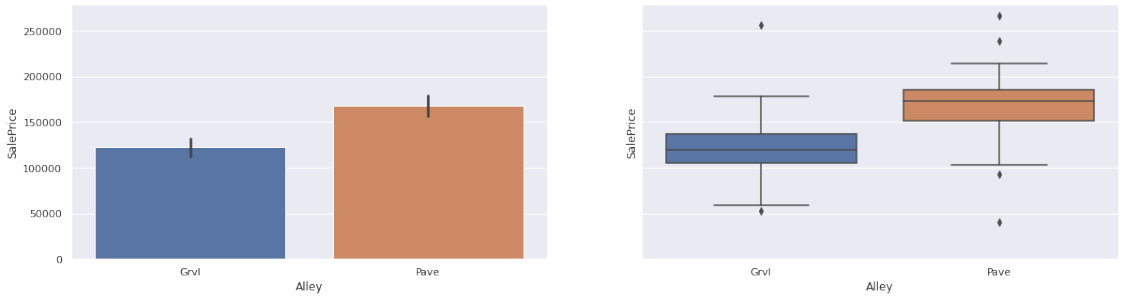
```
continous = ['MSSubClass','YearBuilt' , 'YearRemodAdd' , 'MasVnrArea' ,␣
↪'1stFlrSF' , '2ndFlrSF' , 'LowQualFinSF' , 'GrLivArea' , 'FullBath' ,␣
↪'HalfBath' , 'BedroomAbvGr' , 'KitchenAbvGr' , 'Fireplaces' , 'GarageYrBlt'␣
↪, 'GarageArea' , 'WoodDeckSF' , 'OpenPorchSF' , 'EnclosedPorch' ,␣
↪'3SsnPorch' , 'ScreenPorch' , 'PoolArea' , 'MoSold' ,␣
↪'YrSold','LotFrontage', 'LotArea', 'BsmtFinSF1','BsmtFinSF2','BsmtUnfSF',␣
↪'BsmtFullBath','TotalBsmtSF','BsmtHalfBath','TotRmsAbvGrd','MiscVal']
```

1.5.1 3.5.1 Nominal Features

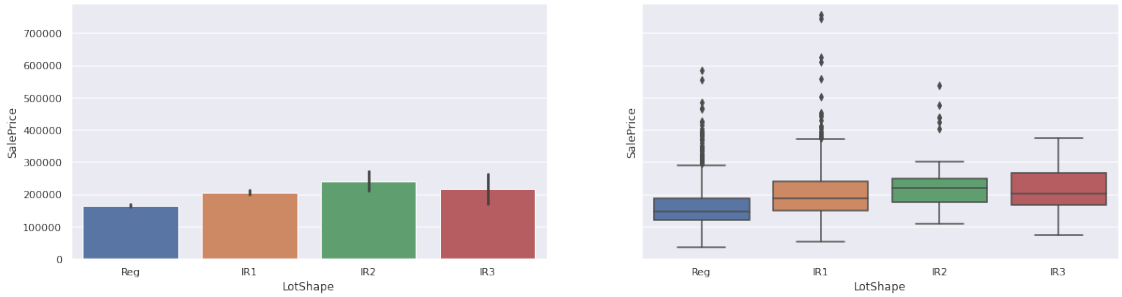
[31]:
```
for feature in nominal:
    fig, ax =plt.subplots(1,2,figsize=(20, 5), sharey=True)
    sns.barplot(data = df, x=feature, y='SalePrice',ax=ax[0])
    sns.boxplot(data=df, x=feature, y="SalePrice", ax=ax[1])
    fig.suptitle(feature)
    fig.show()
```



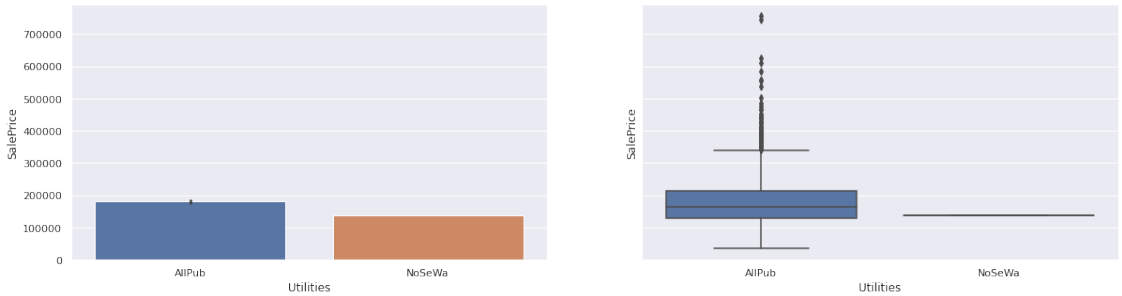
Alley



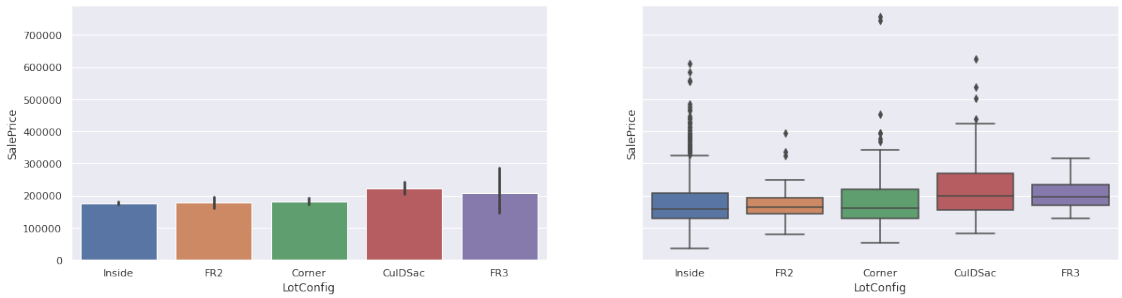
LotShape



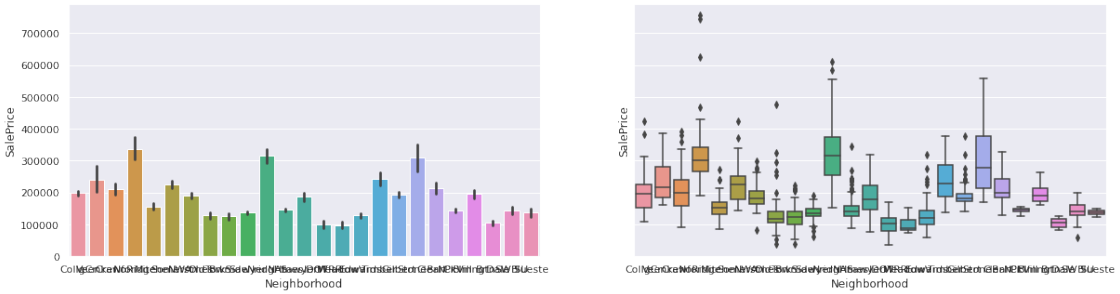
Utilities



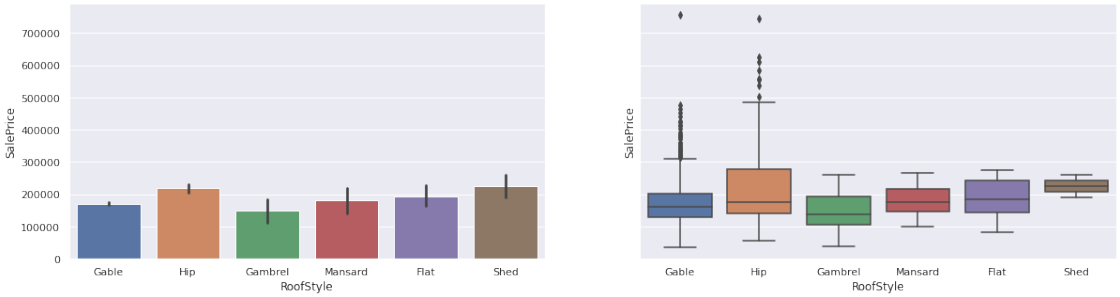
LotConfig



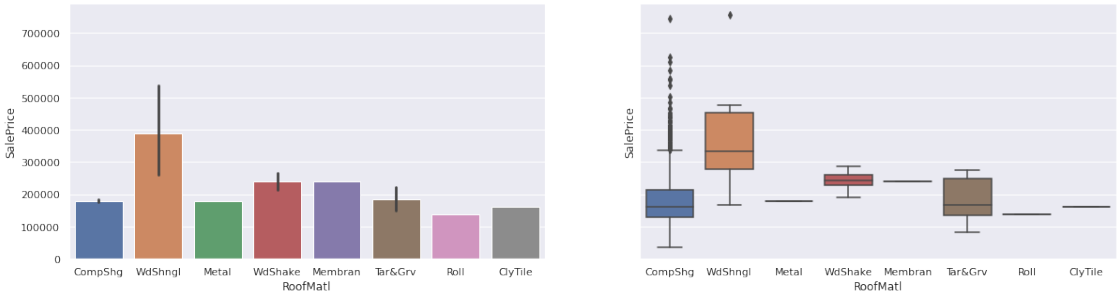
Neighborhood



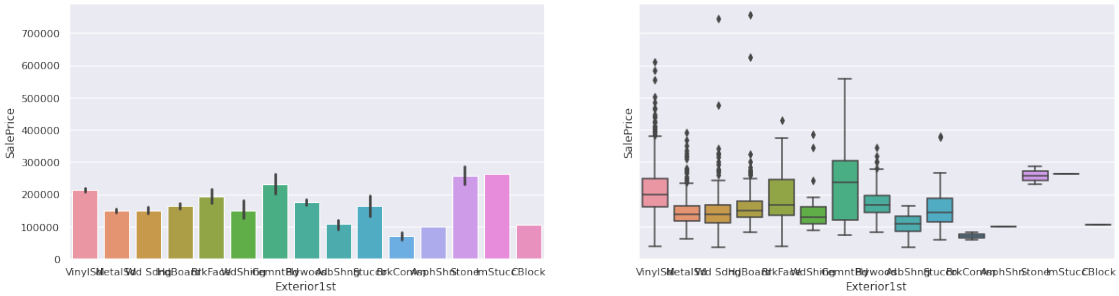
RoofStyle



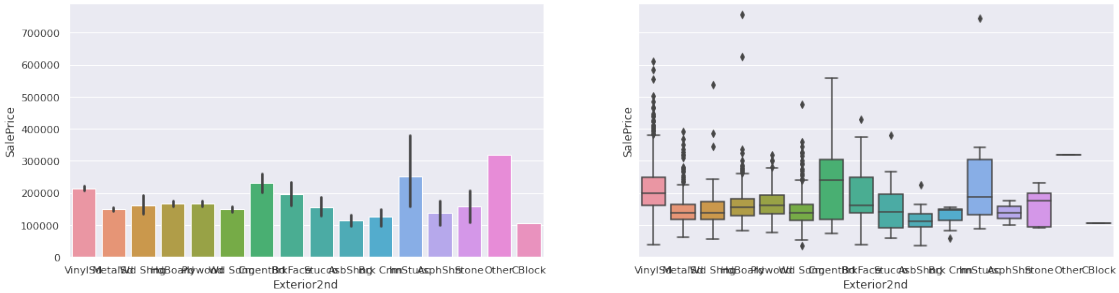
RoofMatl



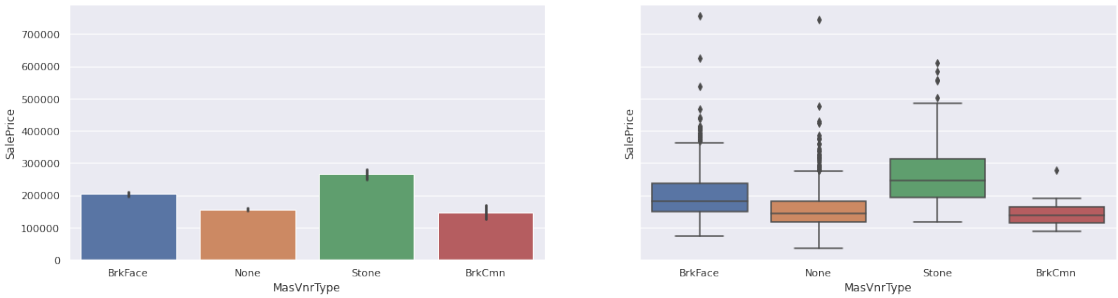
Exterior1st



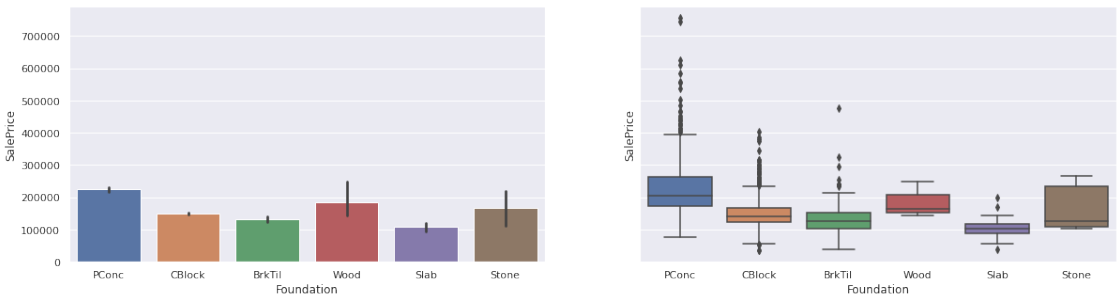
Exterior2nd

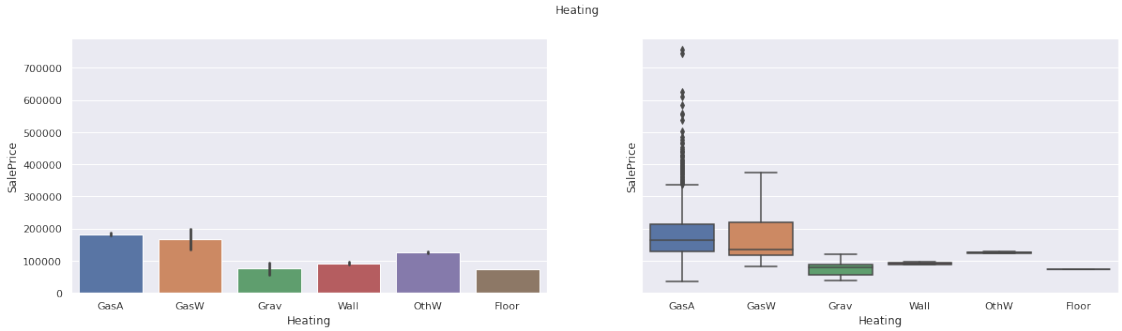


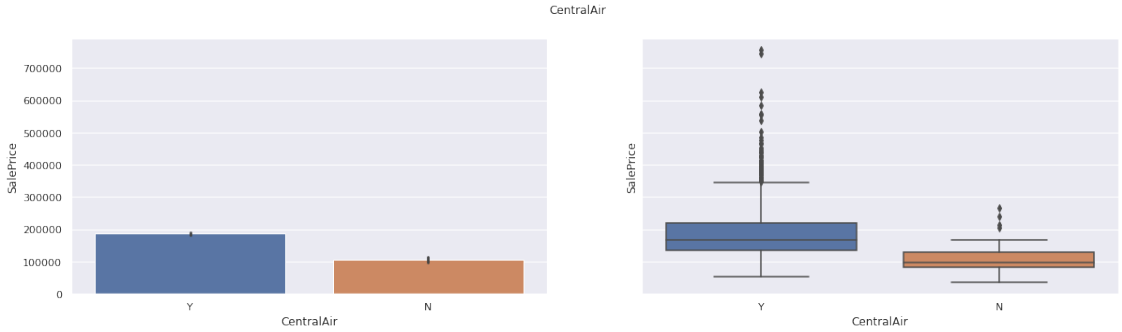
MasVnrType

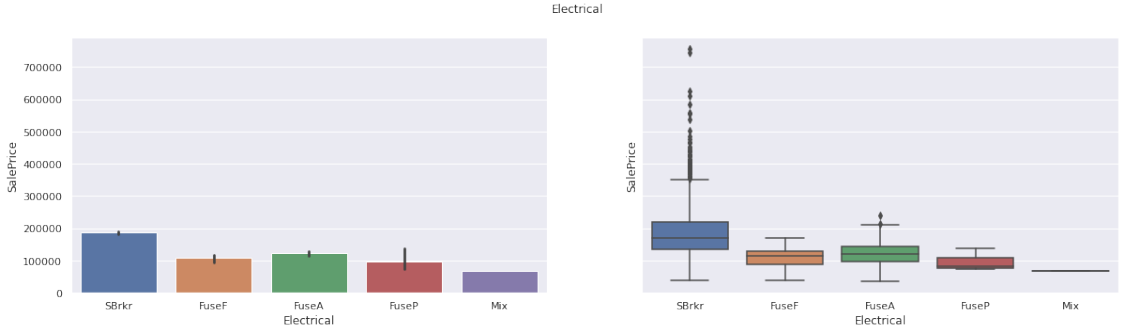


Foundation

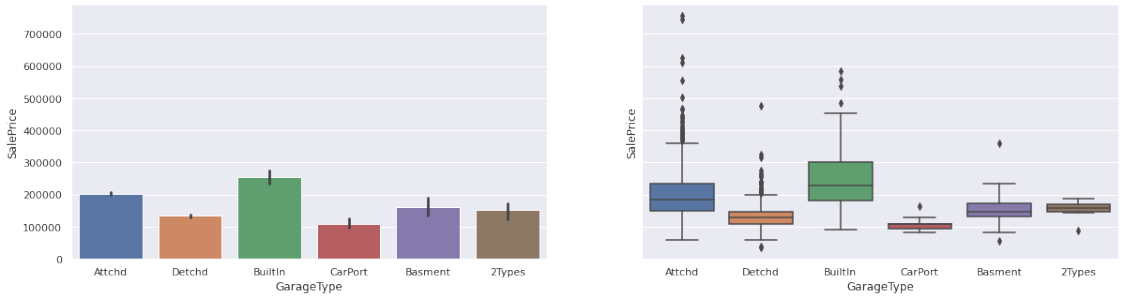



Heating
SalePrice
700000
600000
500000
400000
300000
200000
100000
0
GasA
GasW
Grav
Wall
OthW
Floor
Heating

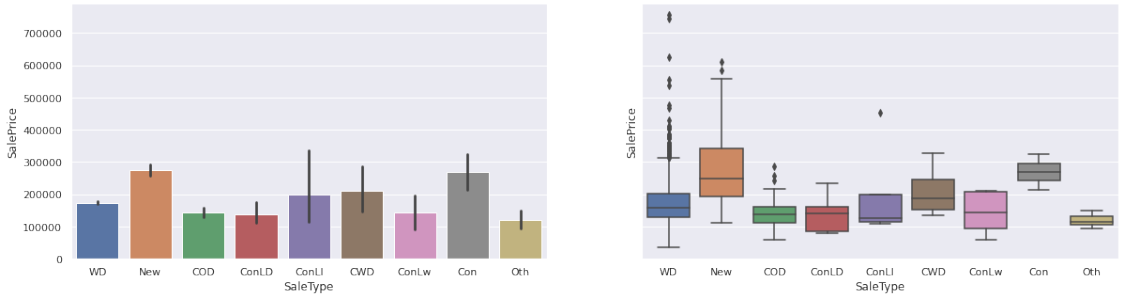

CentralAir
SalePrice
700000
600000
500000
400000
300000
200000
100000
0
Y
N
CentralAir


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

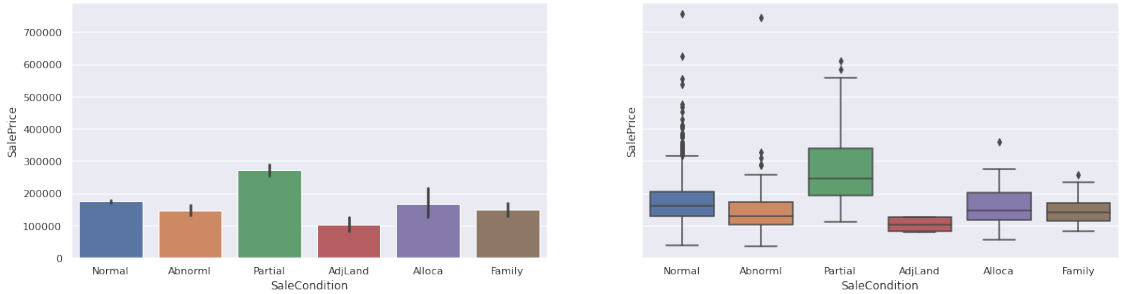
GarageType



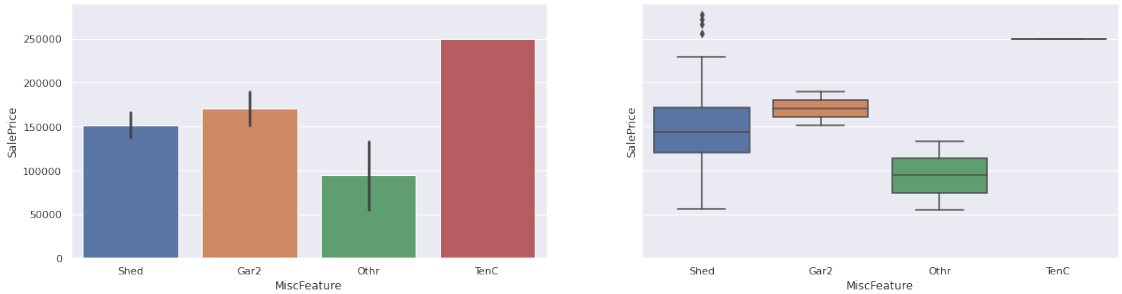
SaleType



SaleCondition

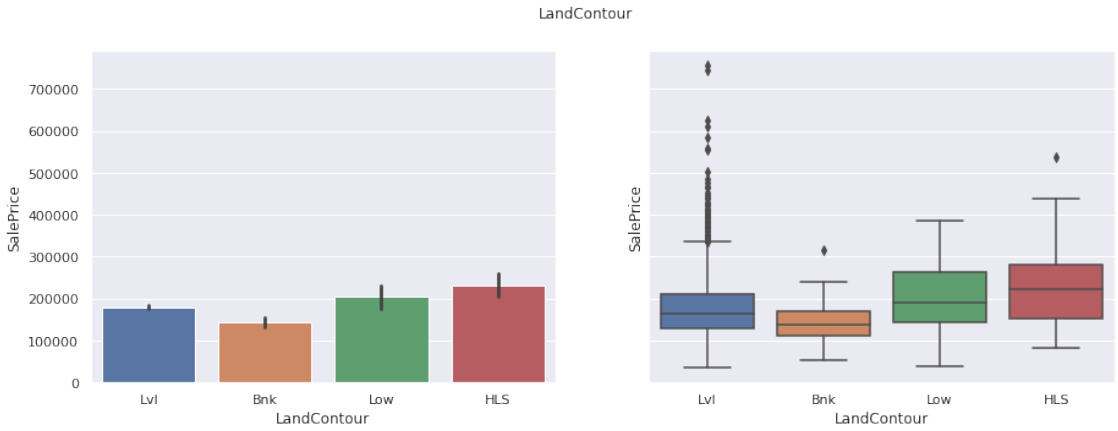


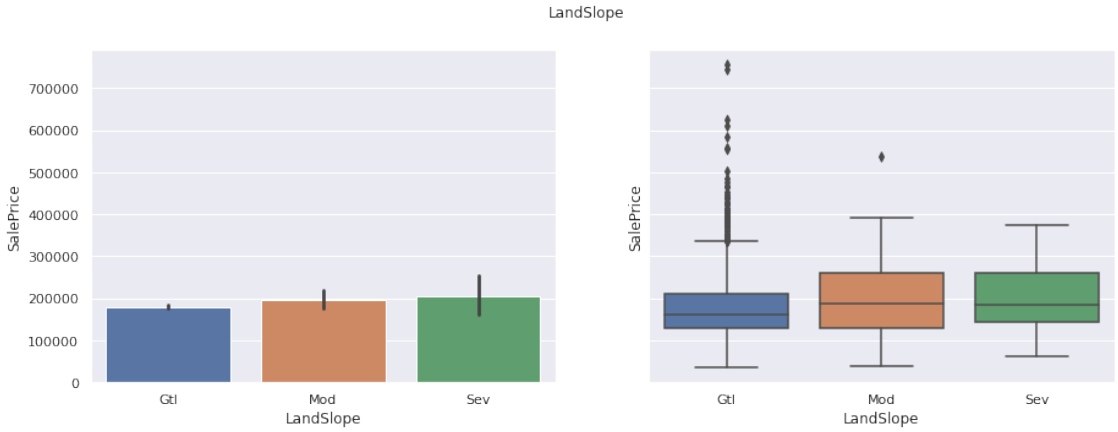
MiscFeature



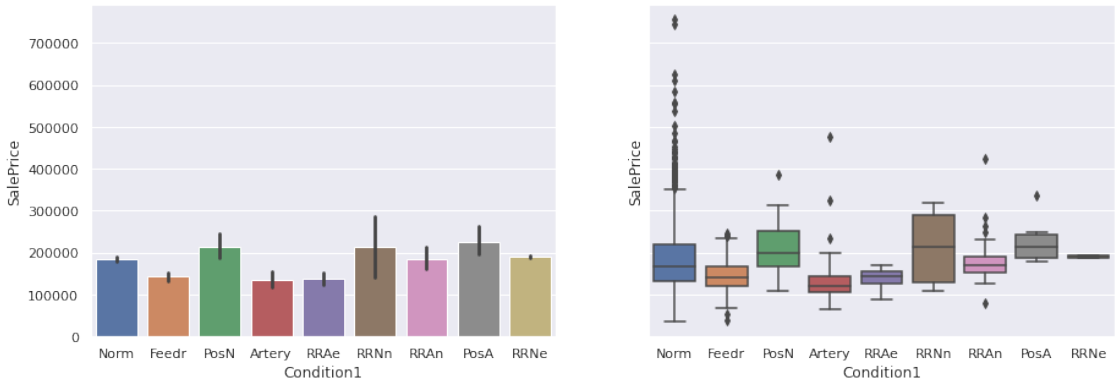
1.5.2 3.5.2 Ordinal Features

[32]:
```
for feature in ordinal:
    fig, ax =plt.subplots(1,2,figsize=(15, 5), sharey=True)
    sns.barplot(data = df, x=feature, y='SalePrice',ax=ax[0])
    sns.boxplot(data=df, x=feature, y="SalePrice", ax=ax[1])
    fig.suptitle(feature)
    fig.show()
```

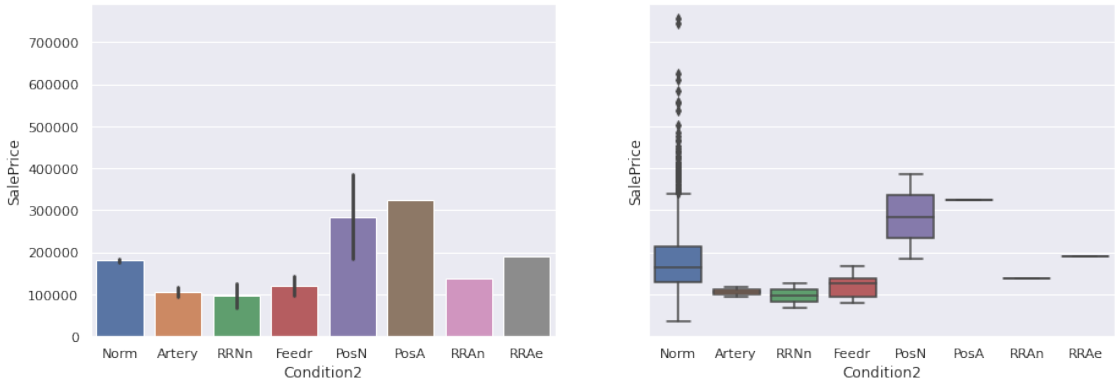




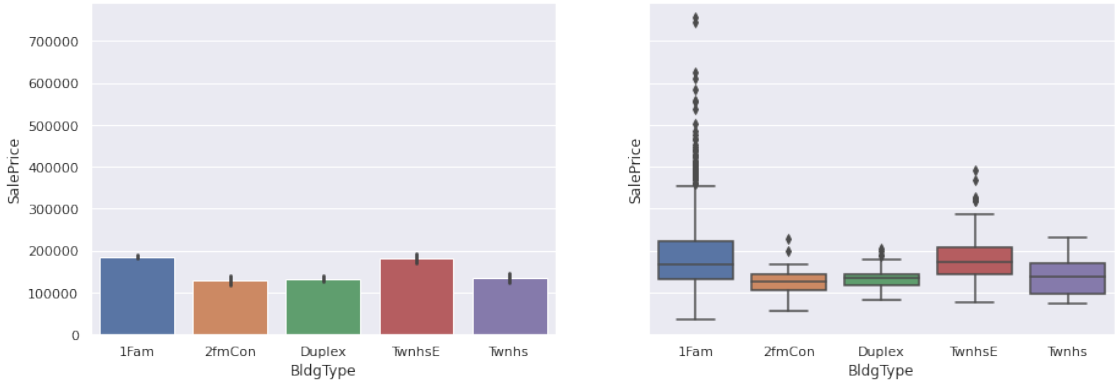
Condition1



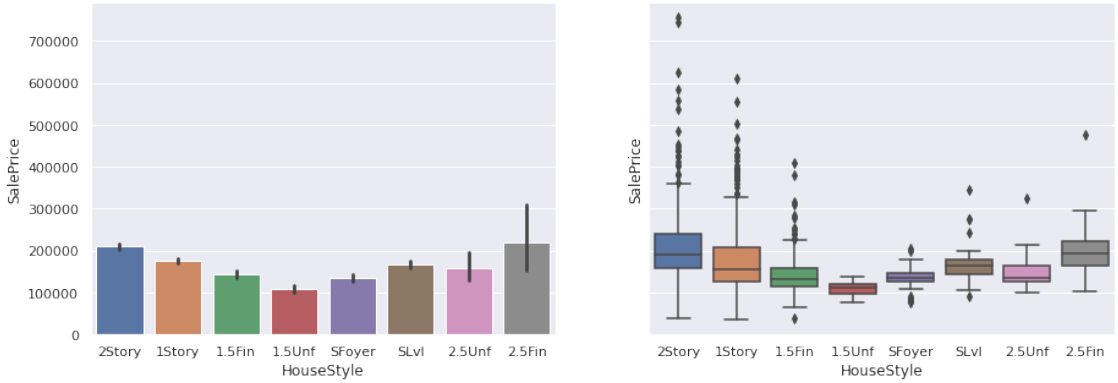
Condition2



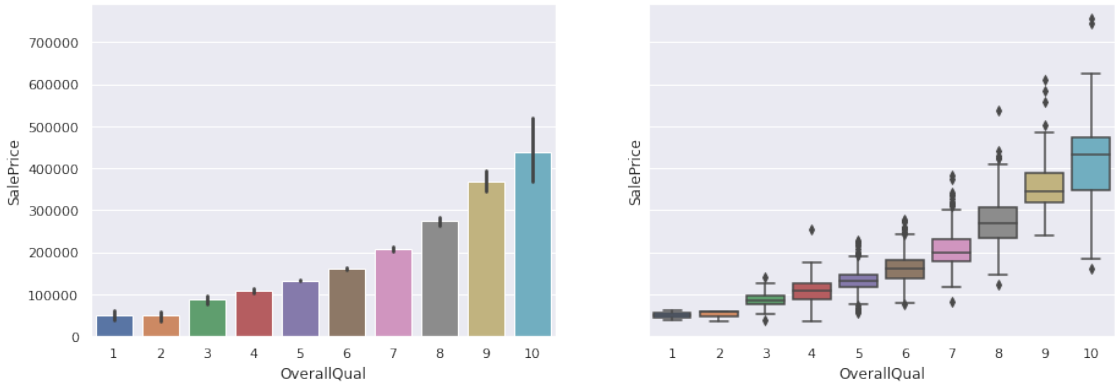
BldgType



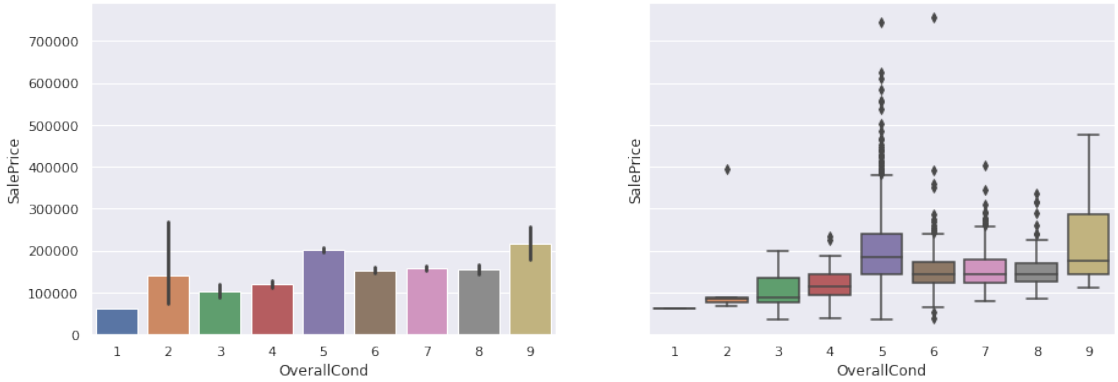
HouseStyle



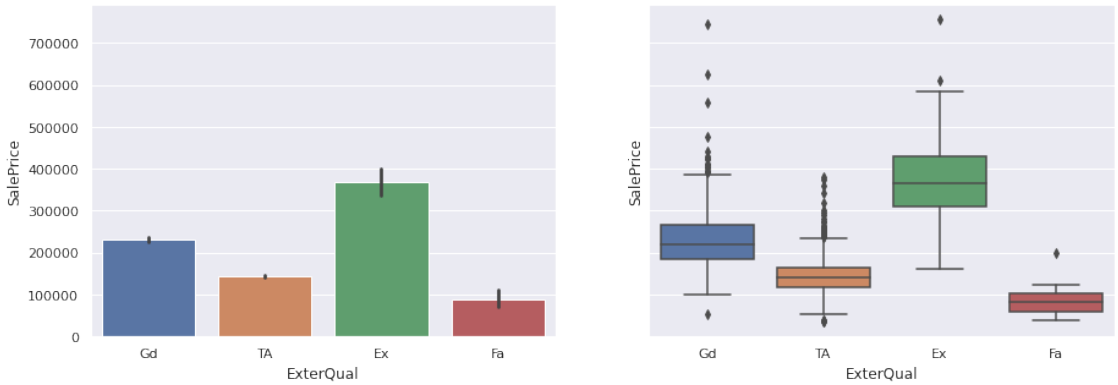
OverallQual



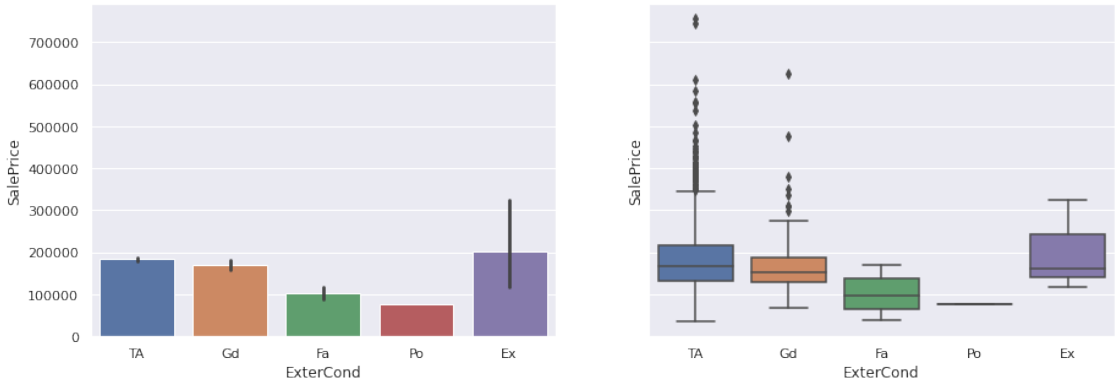
OverallCond



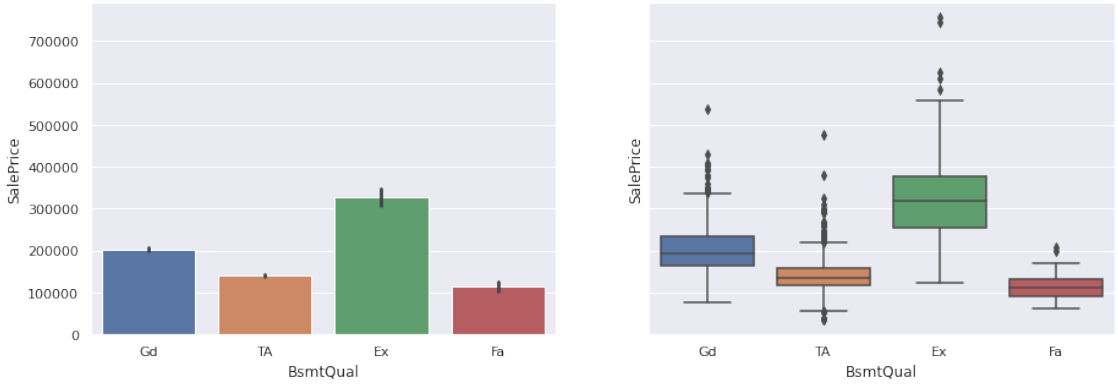
ExterQual



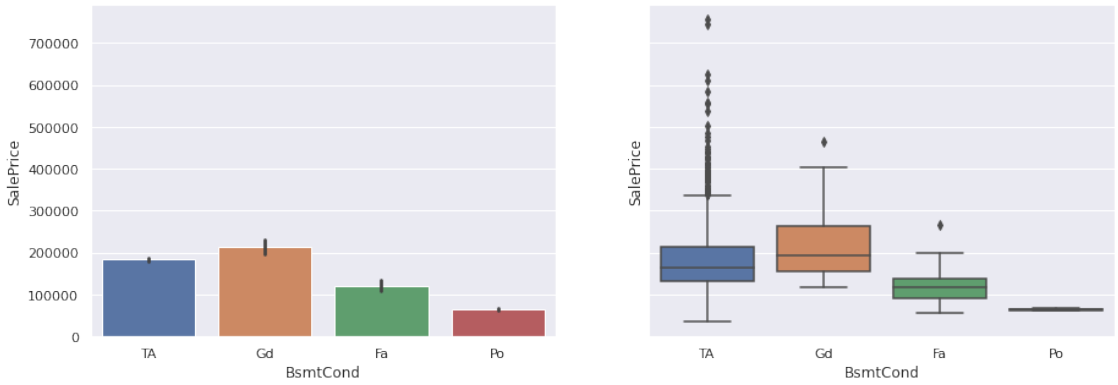
ExterCond



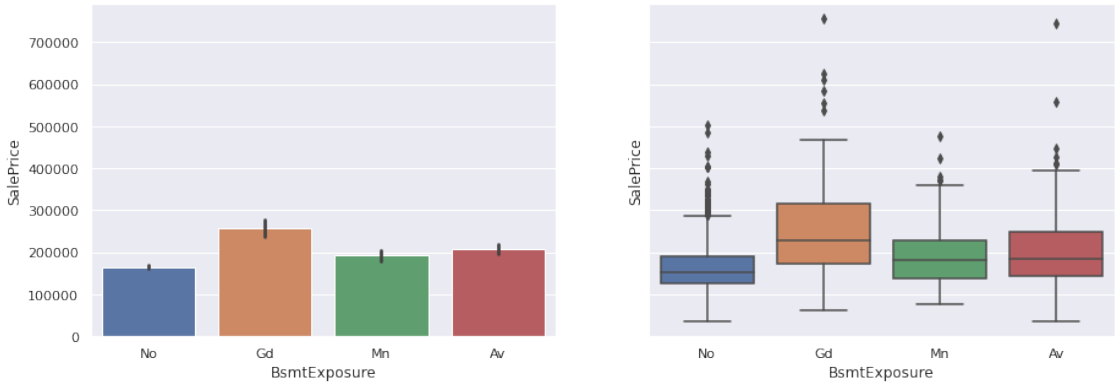
BsmtQual



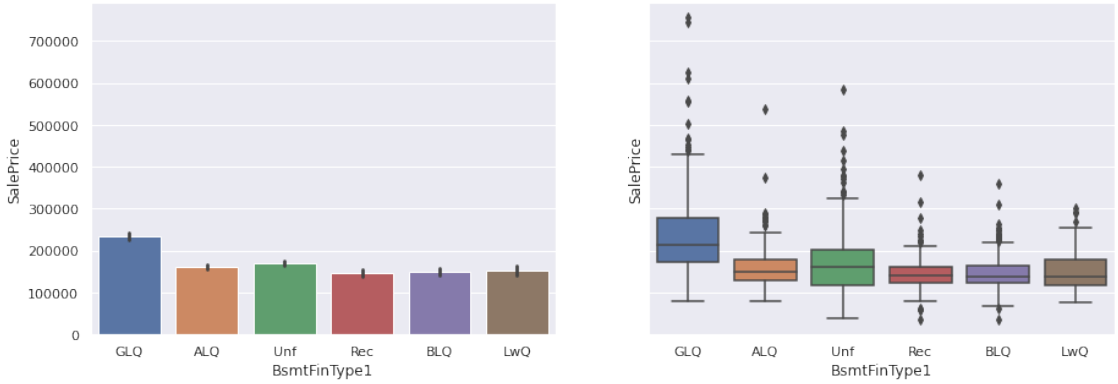
BsmtCond



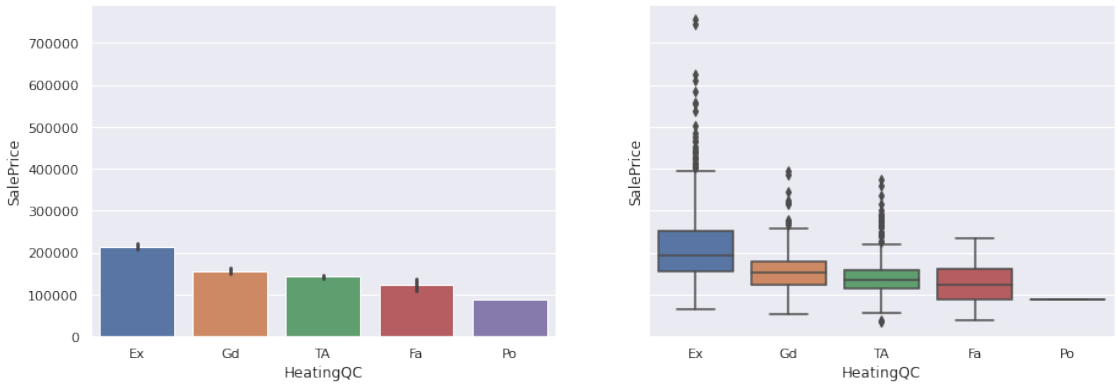
BsmtExposure



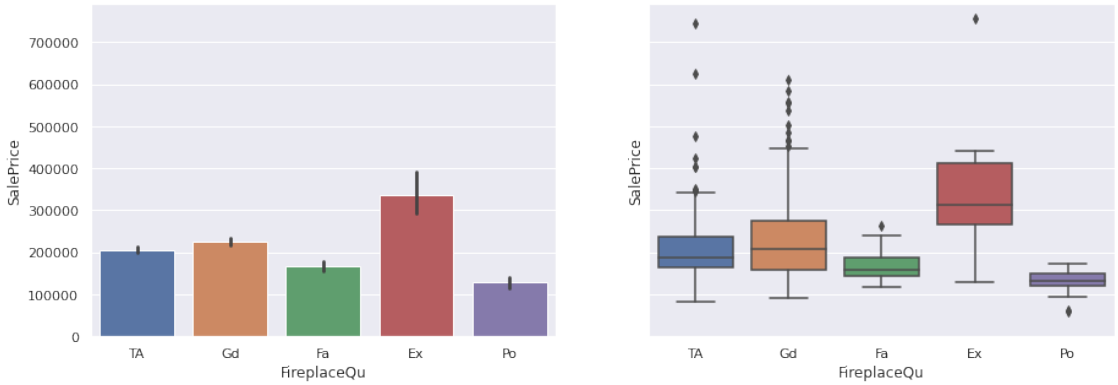
BsmtFinType1



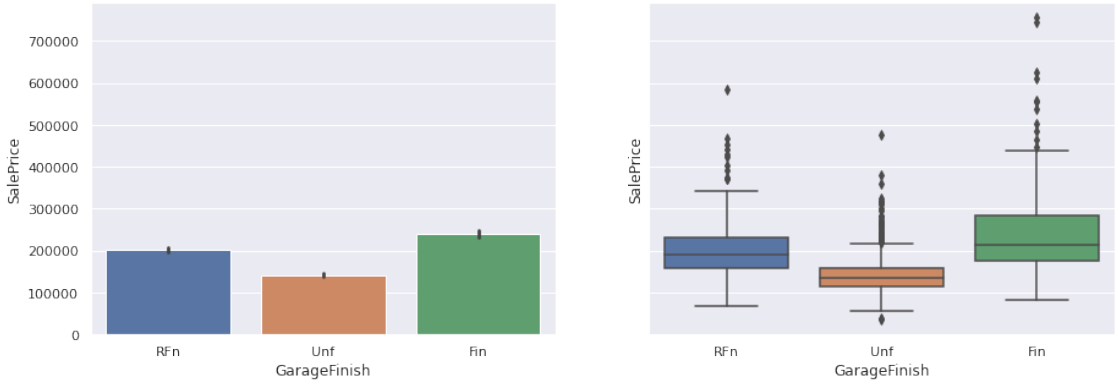
HeatingQC



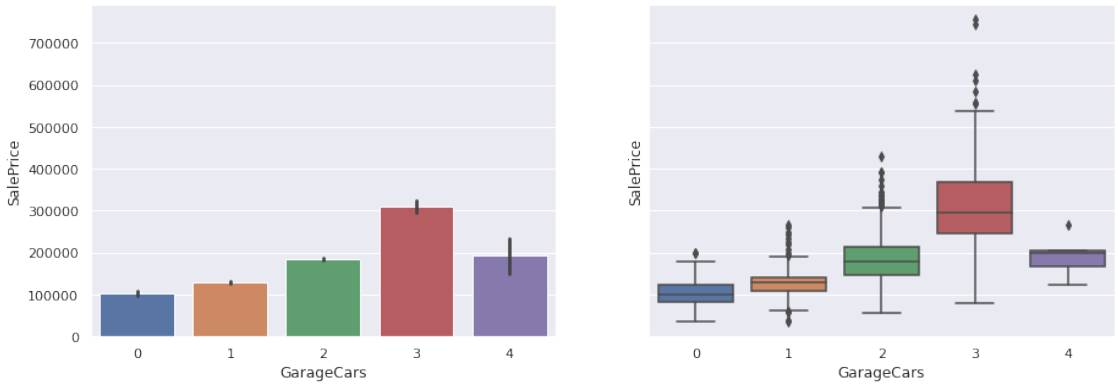
FireplaceQu



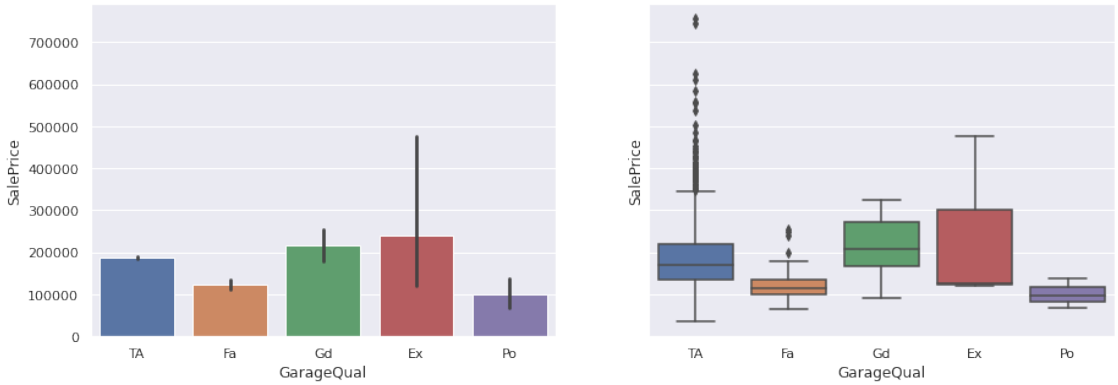
GarageFinish



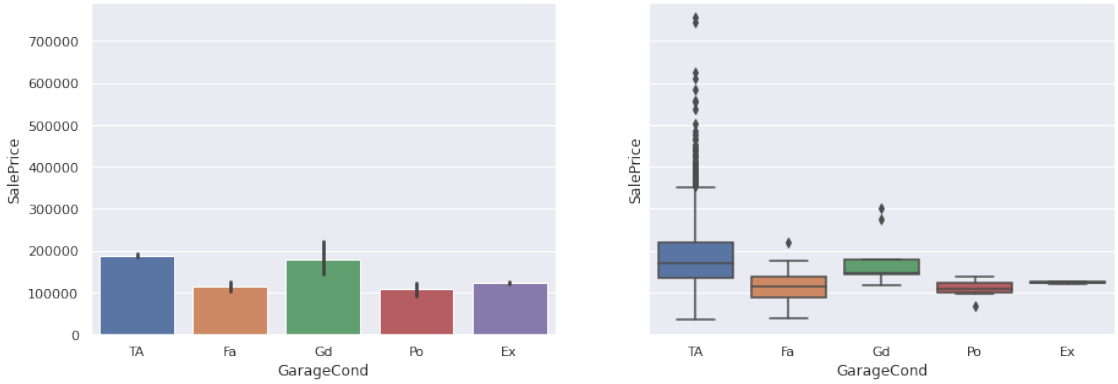
GarageCars



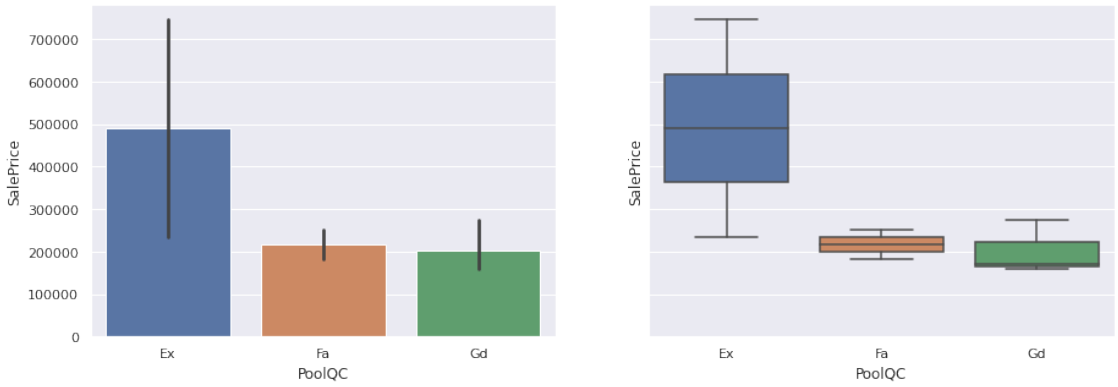
GarageQual



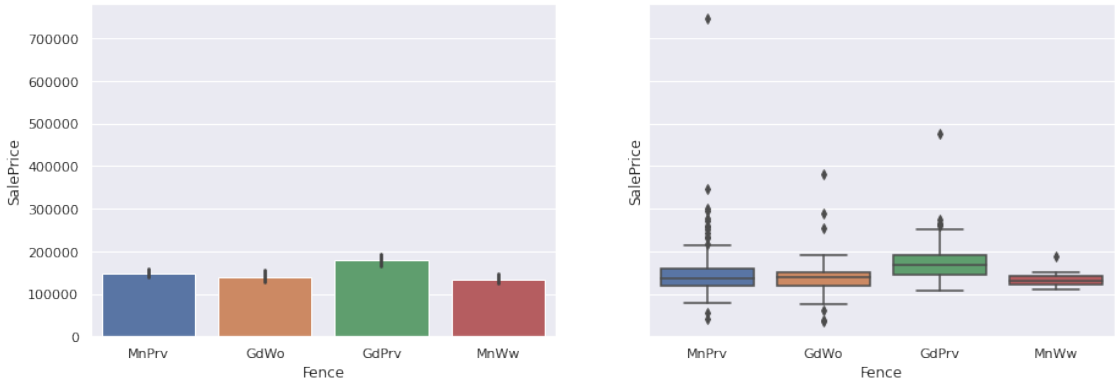
GarageCond



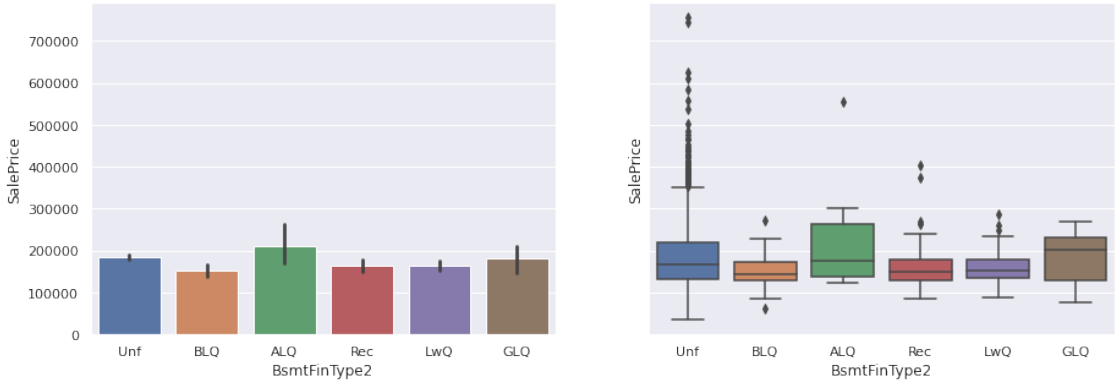
PoolQC



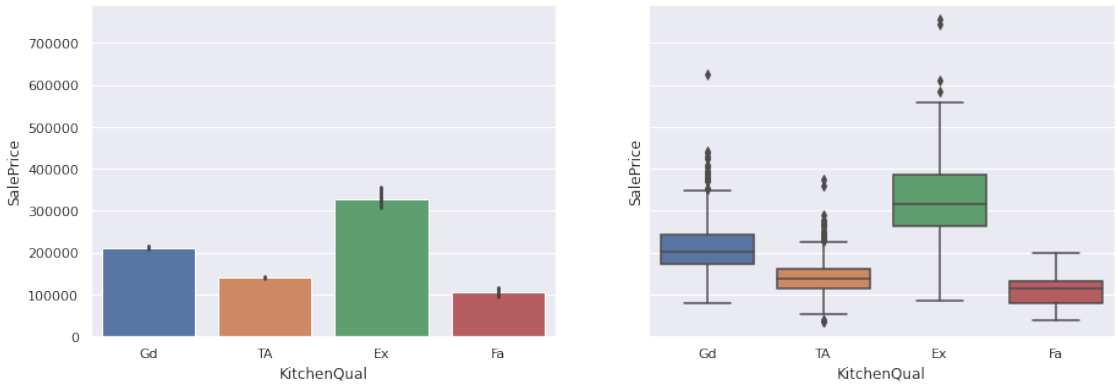
Fence



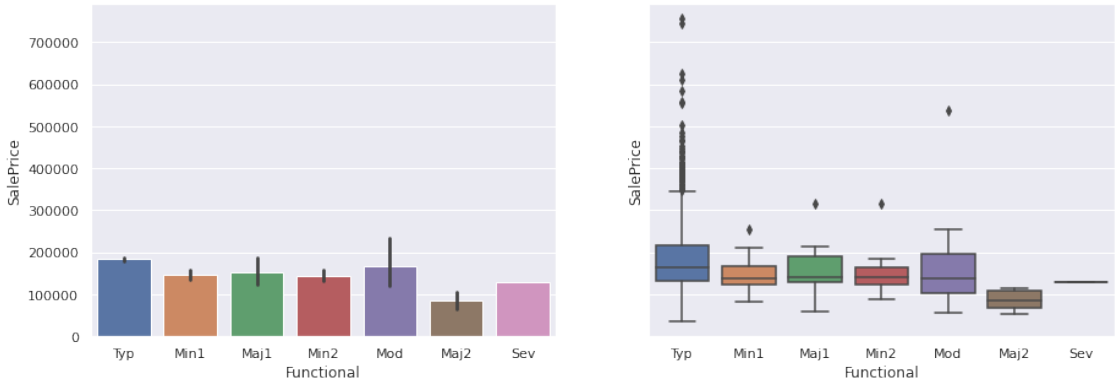
BsmtFinType2



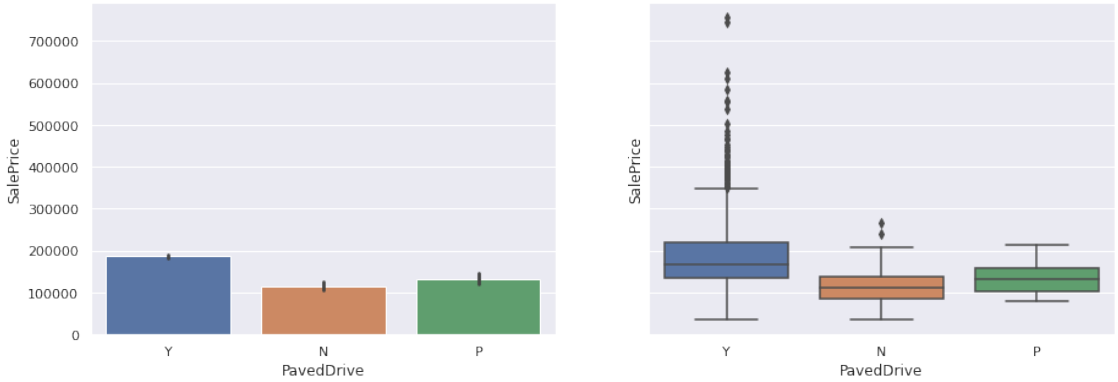
KitchenQual



Functional

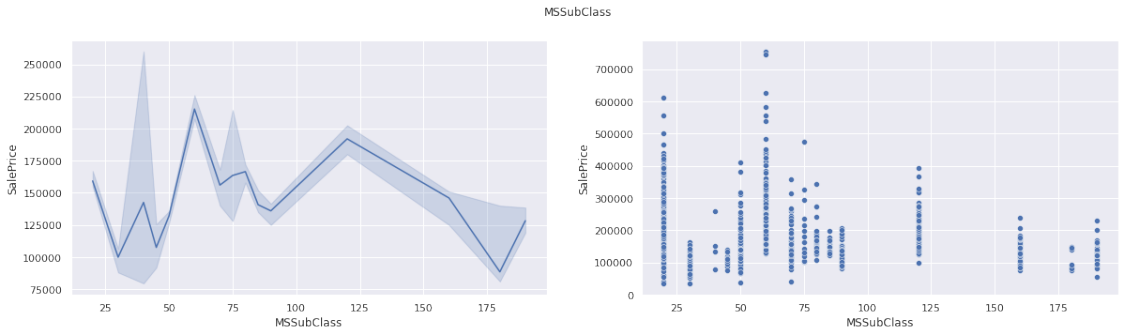


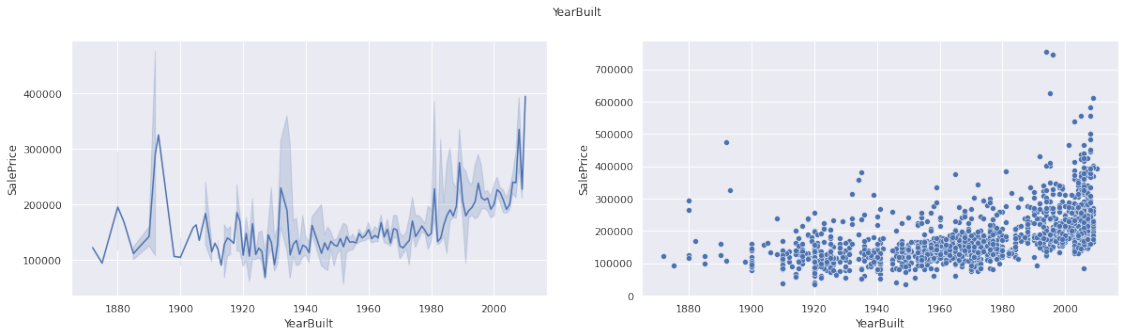
PavedDrive

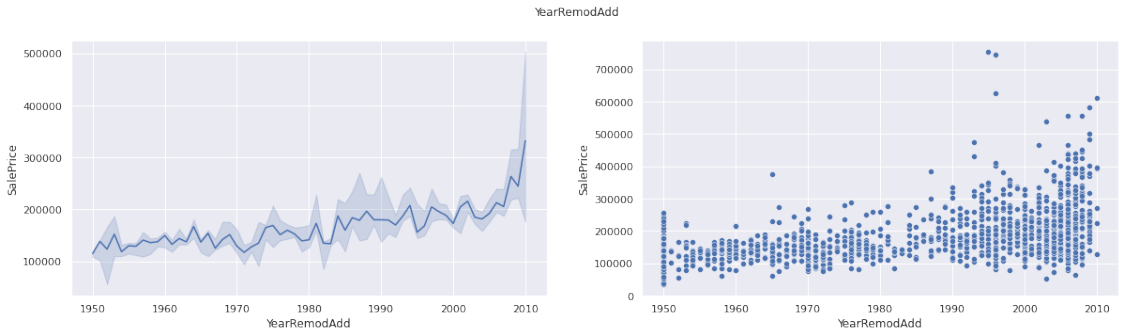


1.5.3 3.5.3 Numerical Features

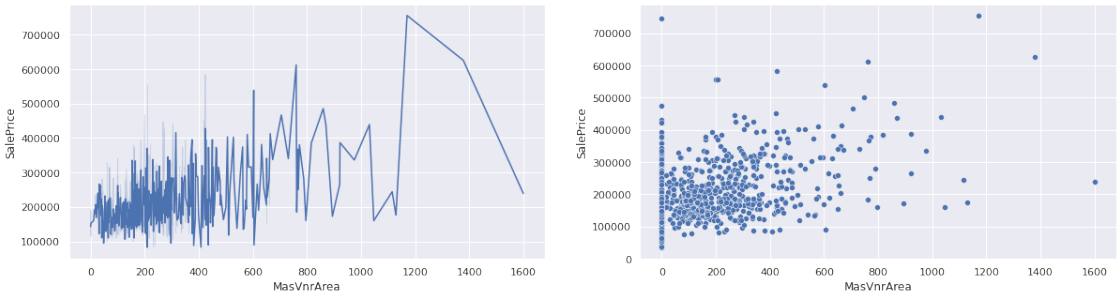
[38]:
```
for feature in continous:
    fig, ax =plt.subplots(1,2,figsize=(20, 5))
    fig.suptitle(feature)
    sns.lineplot(x=feature, y="SalePrice",data=df, estimator=np.median,␣
 ↪ax=ax[0])
    sns.scatterplot(x=feature, y="SalePrice",data=df,  ax=ax[1])
    fig.show()
```



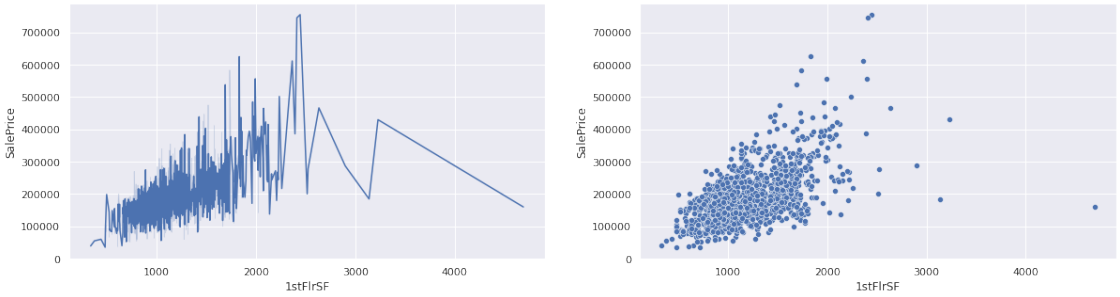




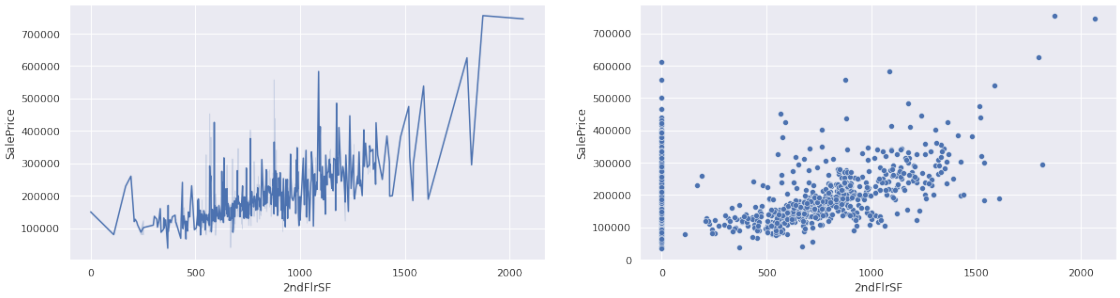
MasVnrArea



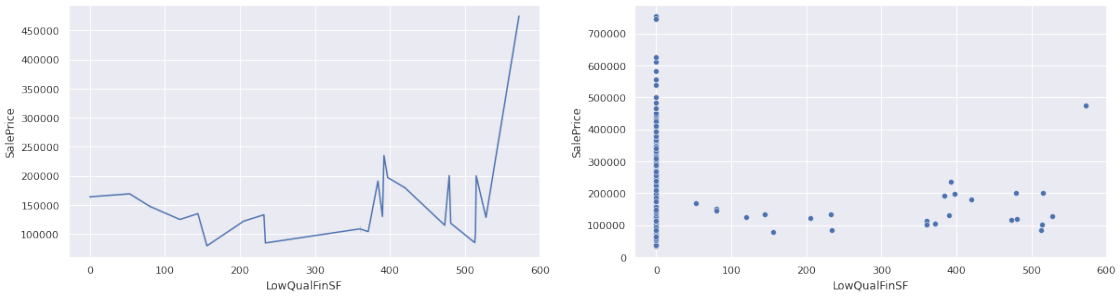
1stFlrSF



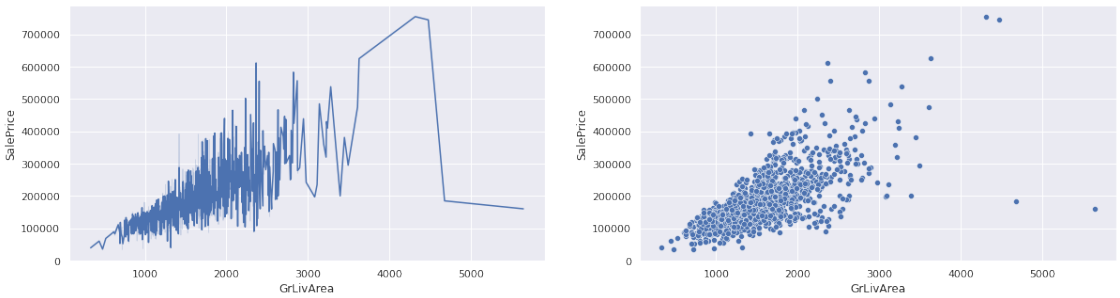
2ndFlrSF



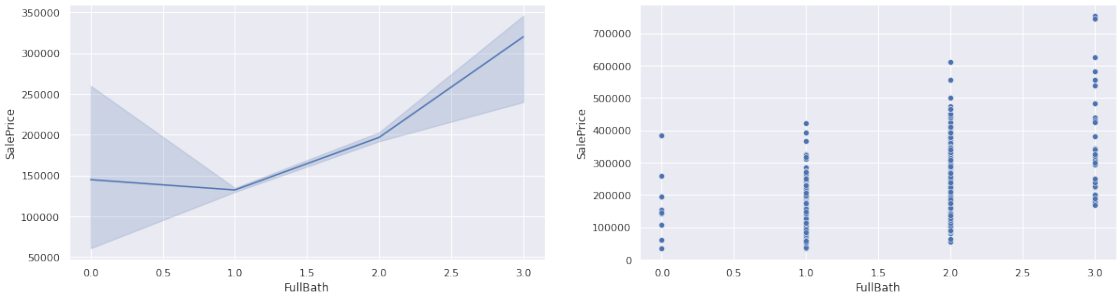
LowQualFinSF



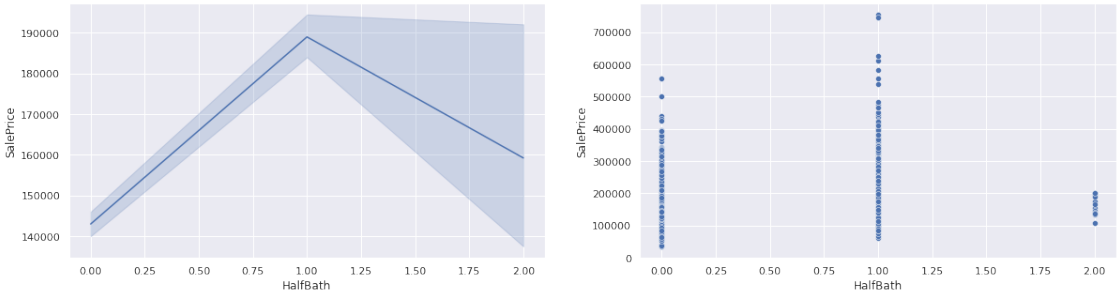
GrLivArea



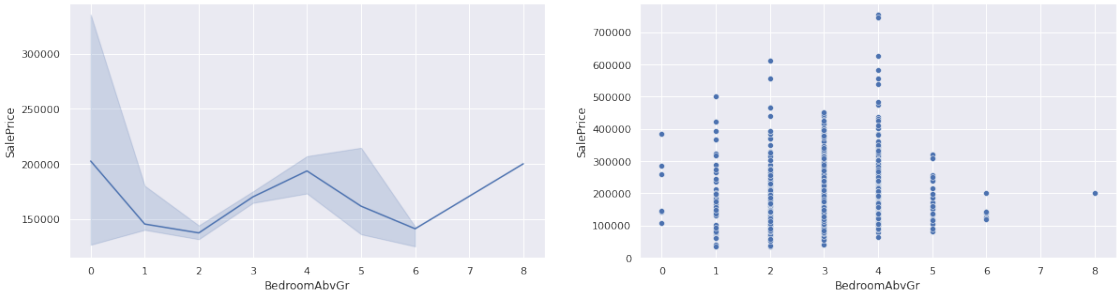
FullBath

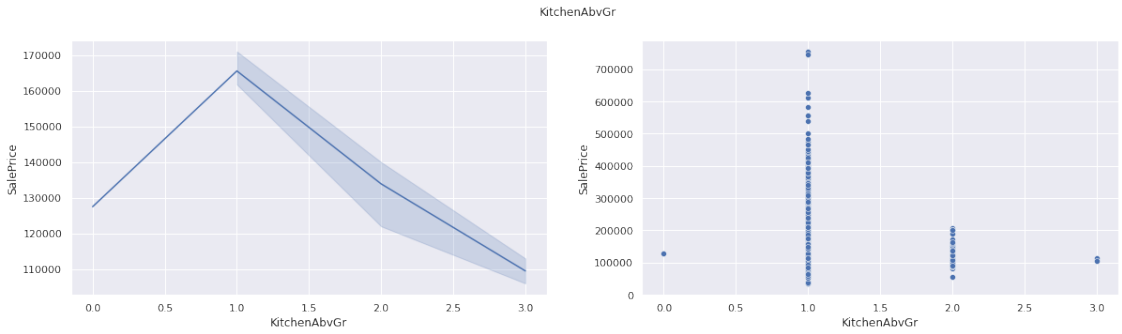


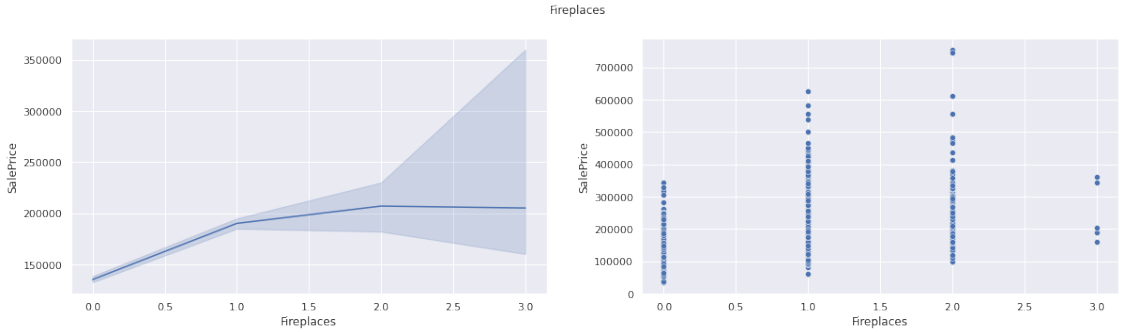
HalfBath

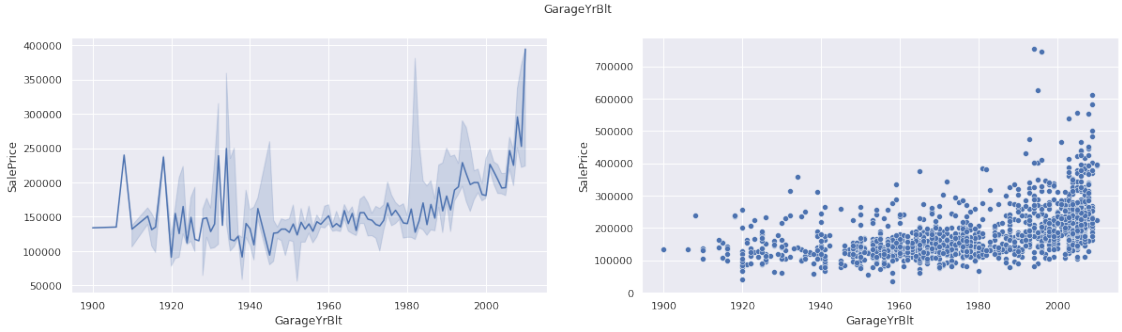


BedroomAbvGr

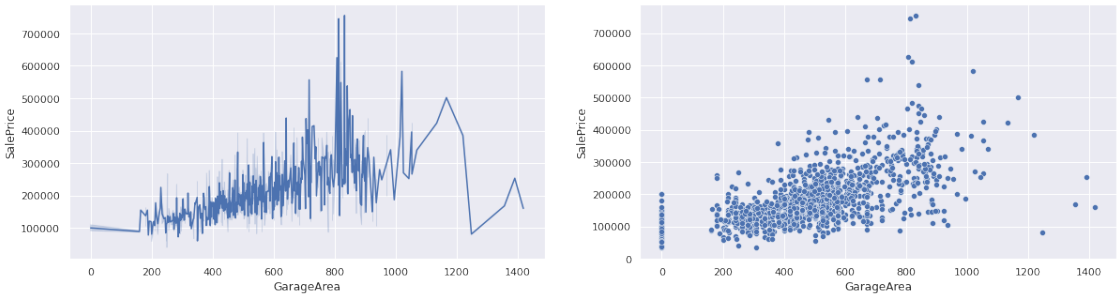



KitchenAbvGr
SalePrice
KitchenAbvGr
SalePrice
KitchenAbvGr

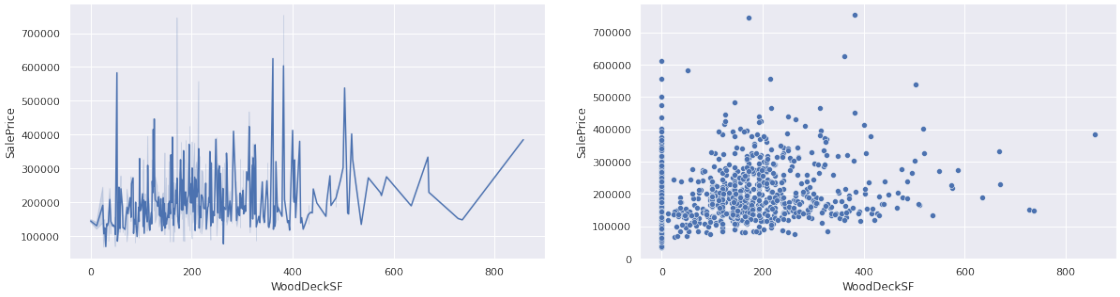

Fireplaces
SalePrice
Fireplaces
SalePrice
Fireplaces


GarageYrBlt
SalePrice
GarageYrBlt
SalePrice
GarageYrBlt

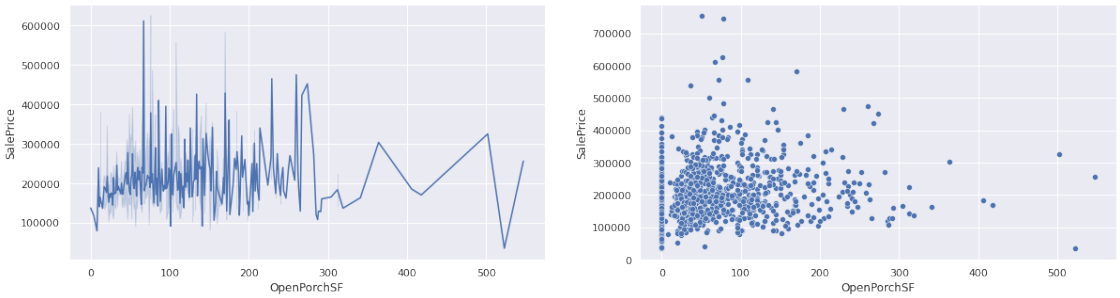
GarageArea



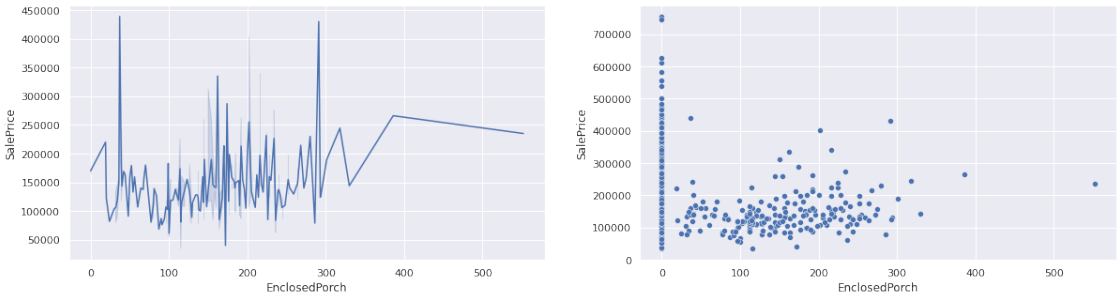
WoodDeckSF



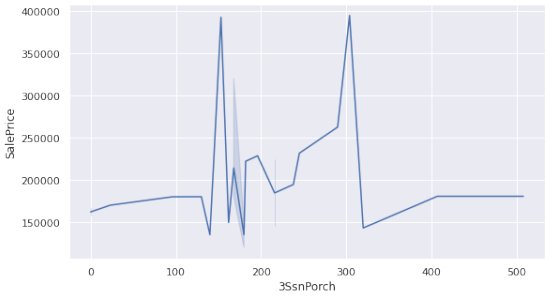
OpenPorchSF

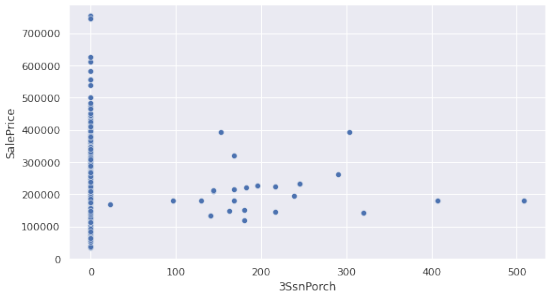


EnclosedPorch

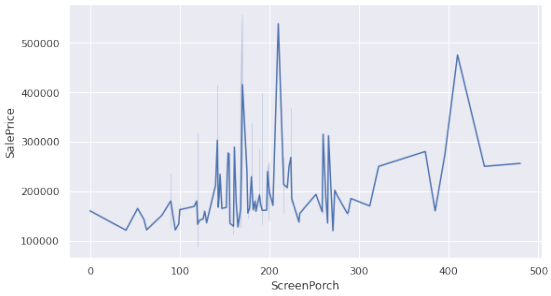


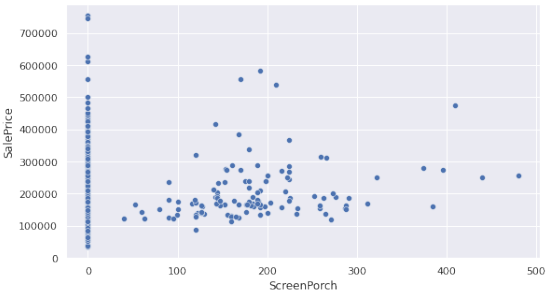
3SsnPorch



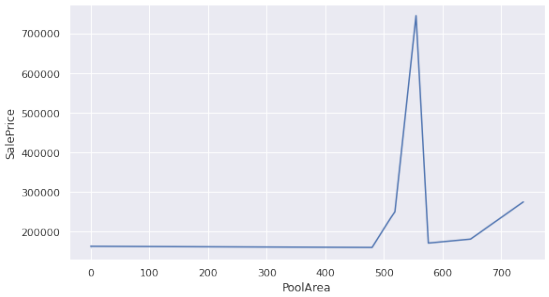


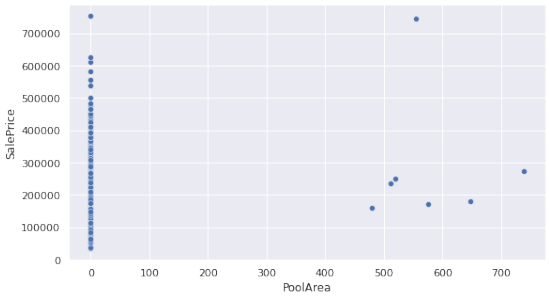
ScreenPorch



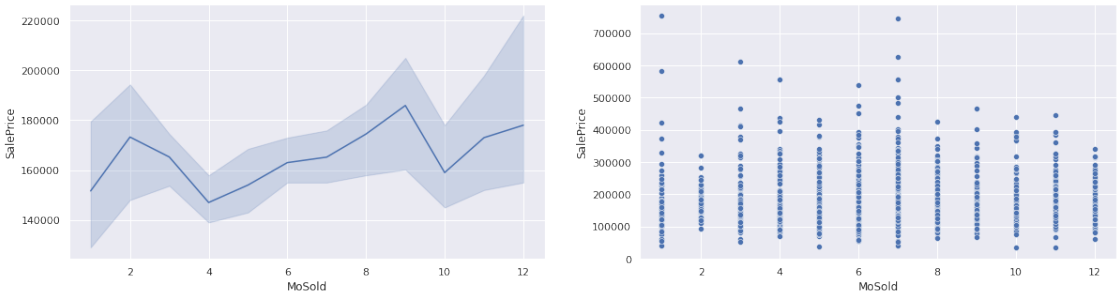


PoolArea

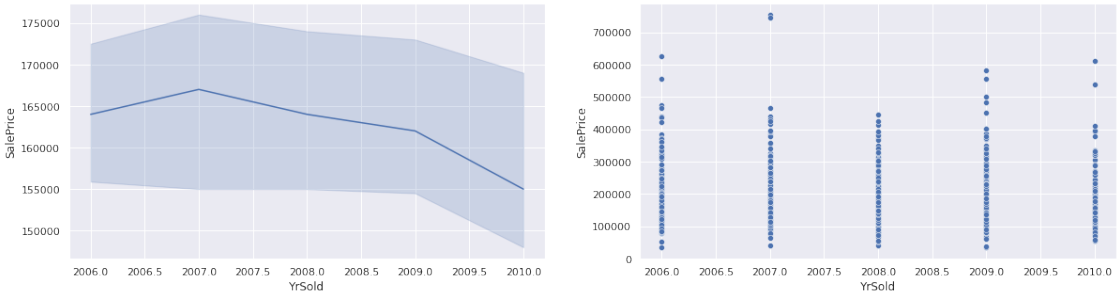




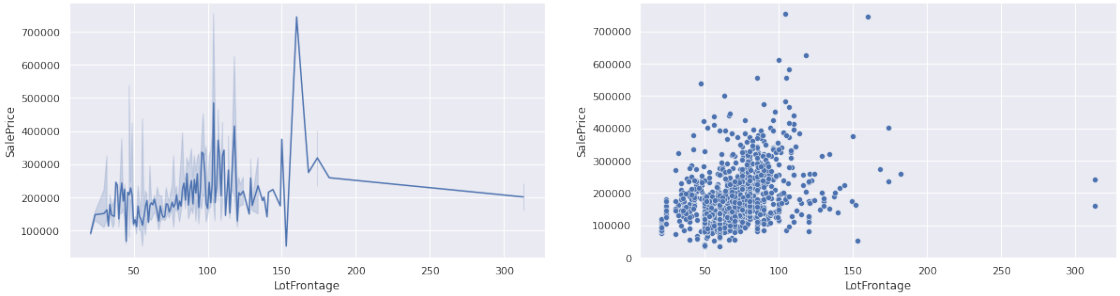
MoSold



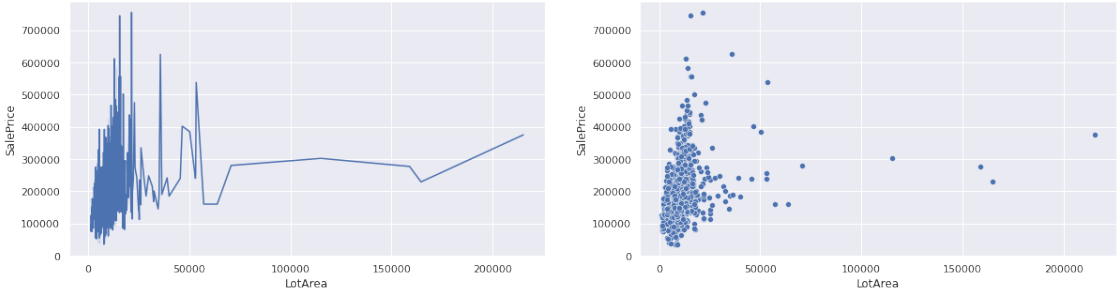
YrSold



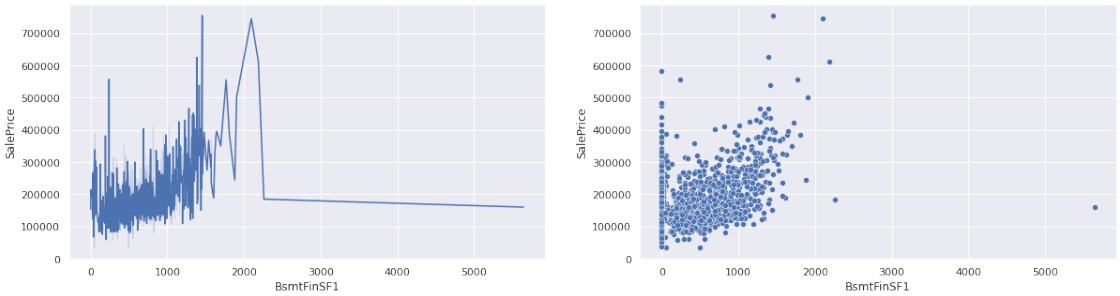
LotFrontage



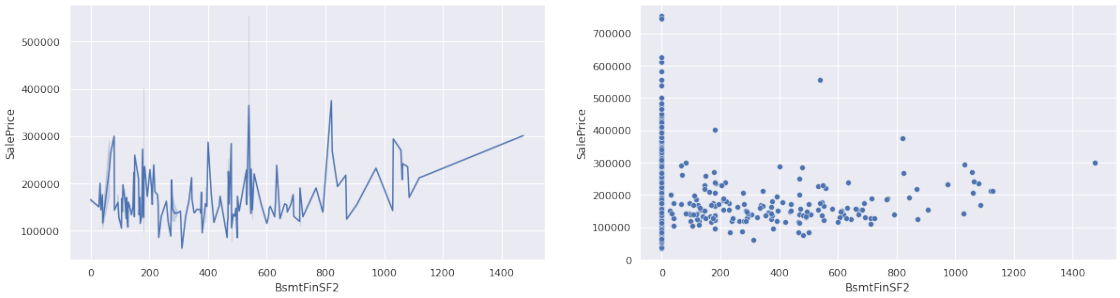
LotArea



BsmtFinSF1

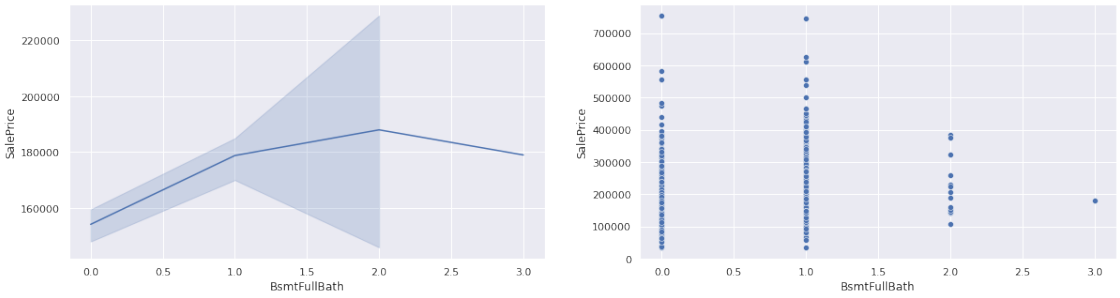


BsmtFinSF2

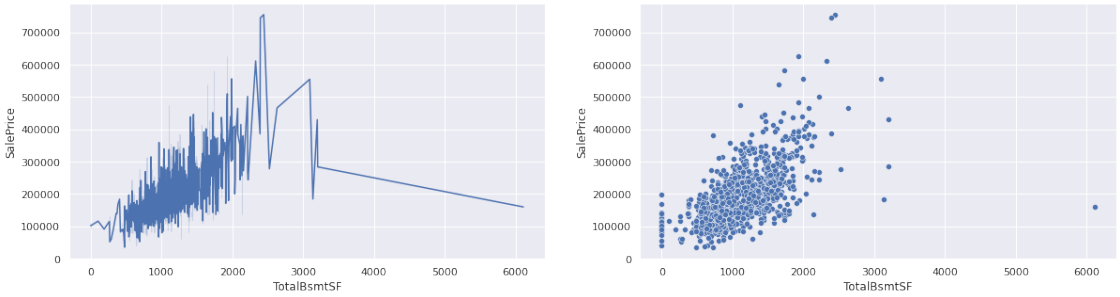


BsmtUnfSF

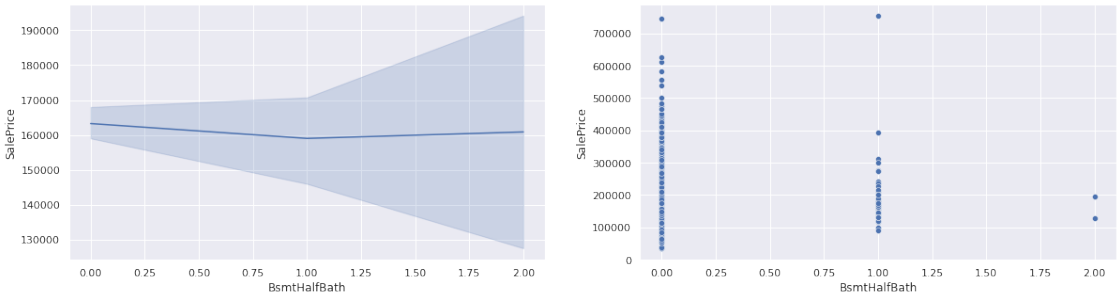
BsmtFullBath



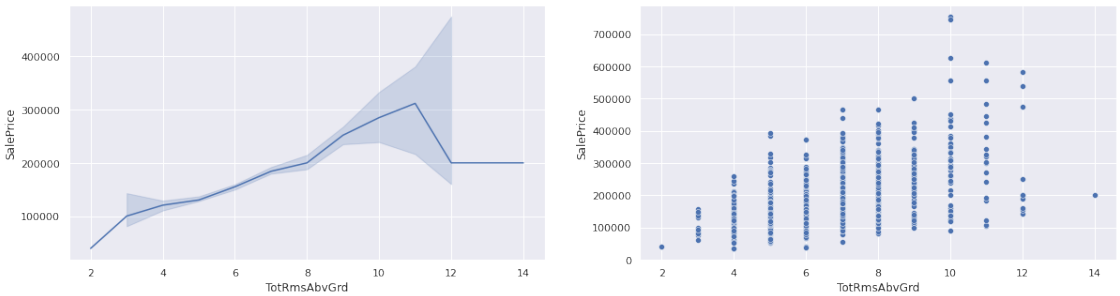
TotalBsmtSF

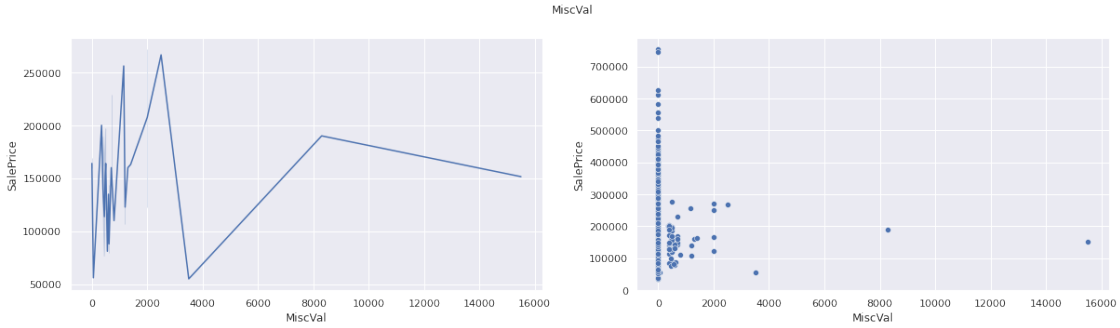


BsmtHalfBath



TotRmsAbvGrd

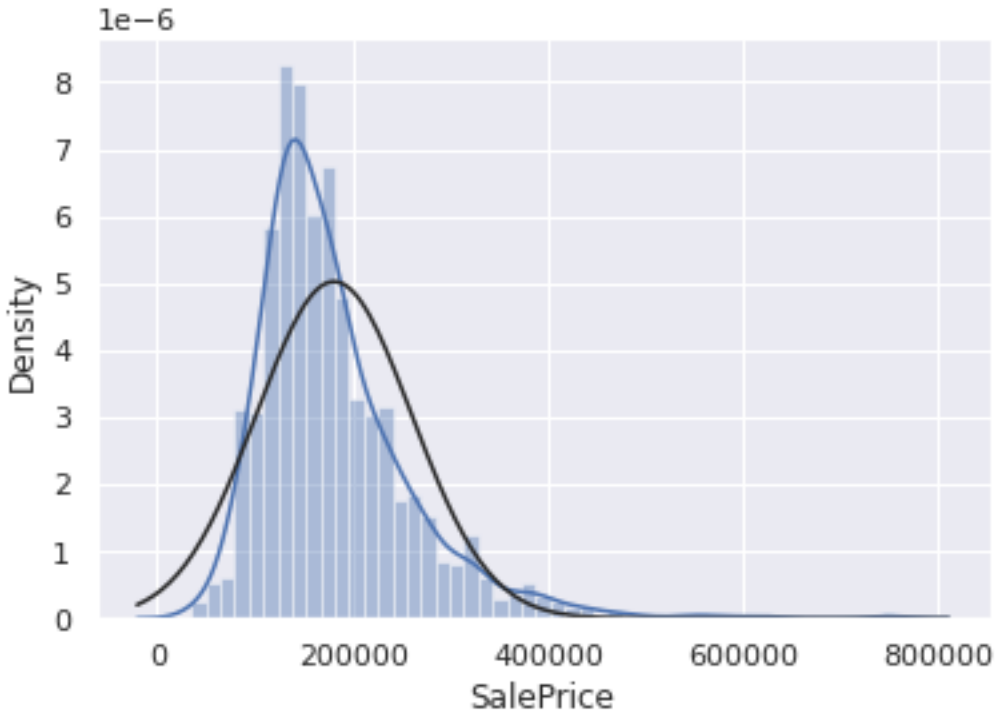


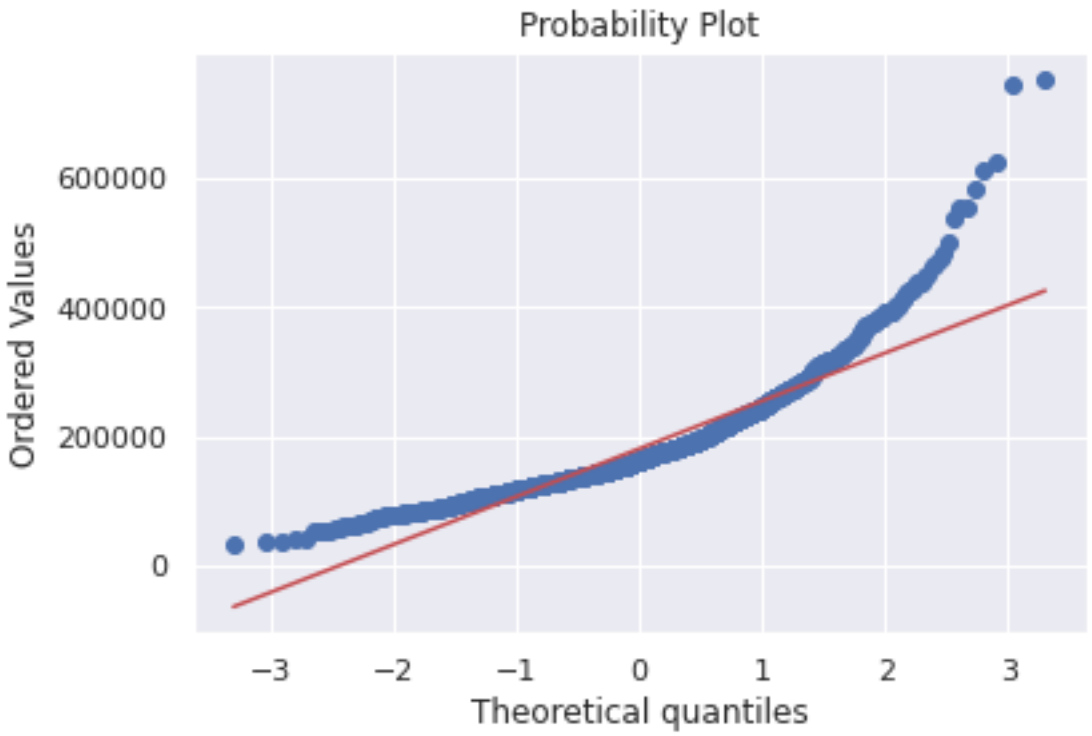


1.6 3.6 Distribution Plots

1.6.1 3.6.1 Sale Price

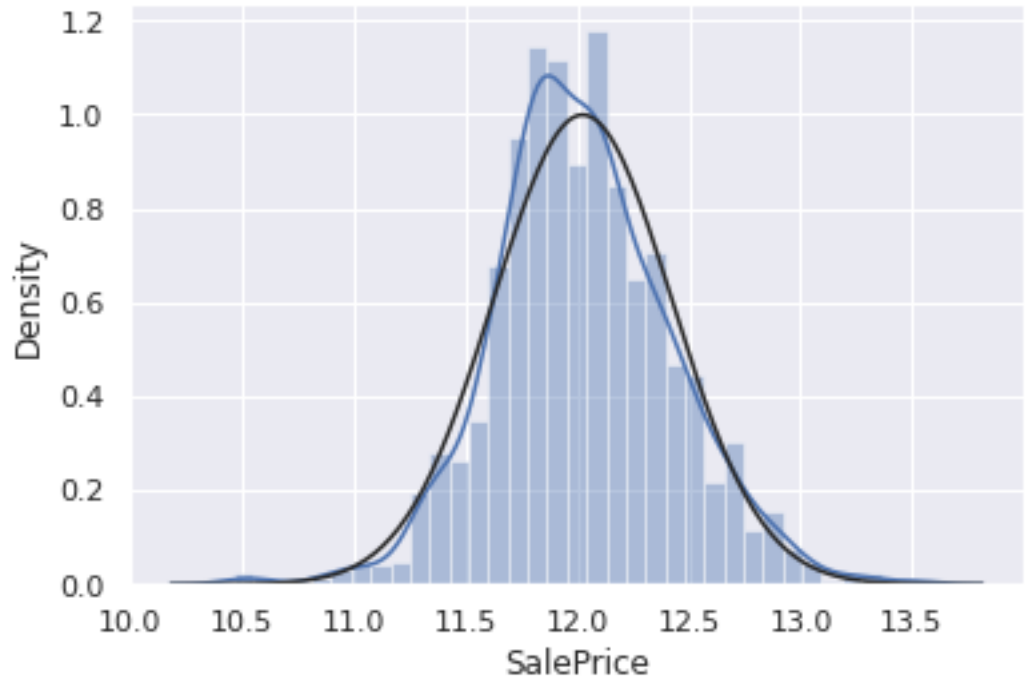
[17]:
```
#histogram and normal probability plot
sns.distplot(df['SalePrice'], fit=norm,);
fig = plt.figure()
res = stats.probplot(df['SalePrice'], plot=plt)
```

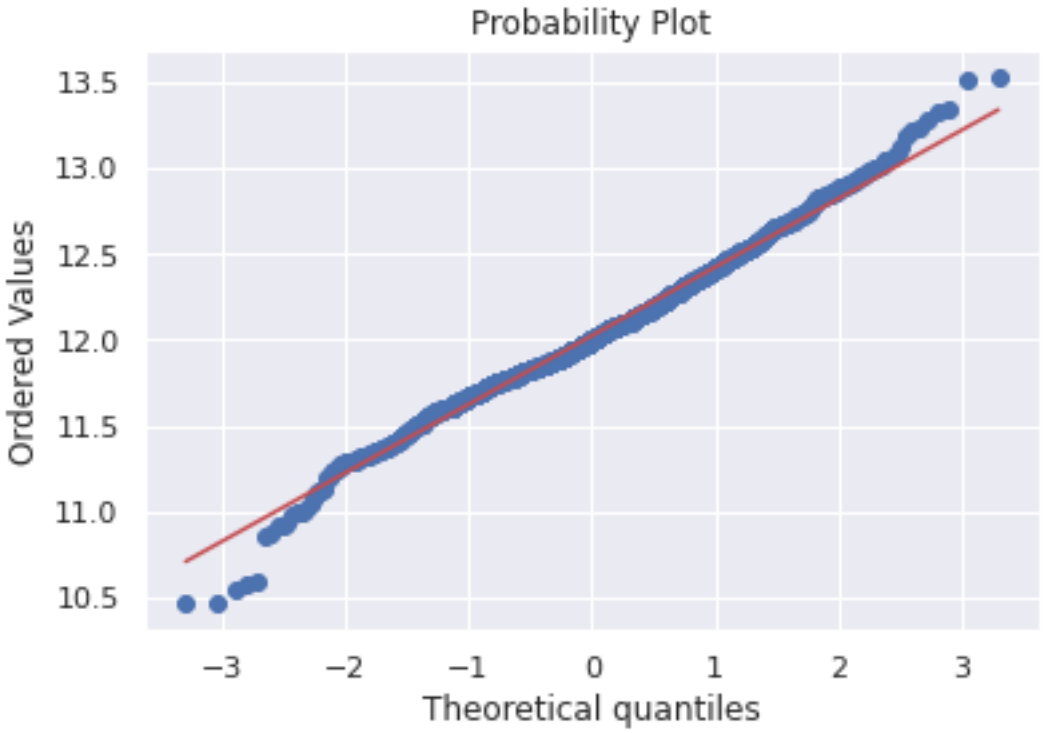




'SalePrice' is not normal. It shows 'peakedness', positive skewness and does not follow the diagonal line. We can log transform this feature to get normal distribution.

[18]:
```
# Applying log transformation
df['SalePrice'] = np.log(df['SalePrice'])

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

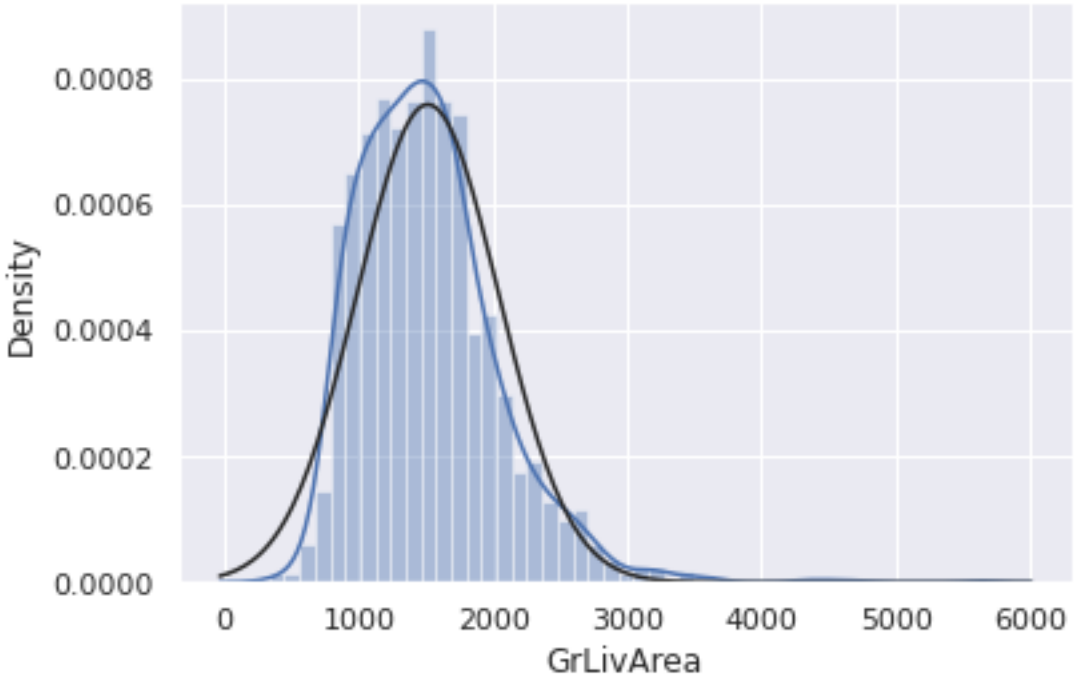


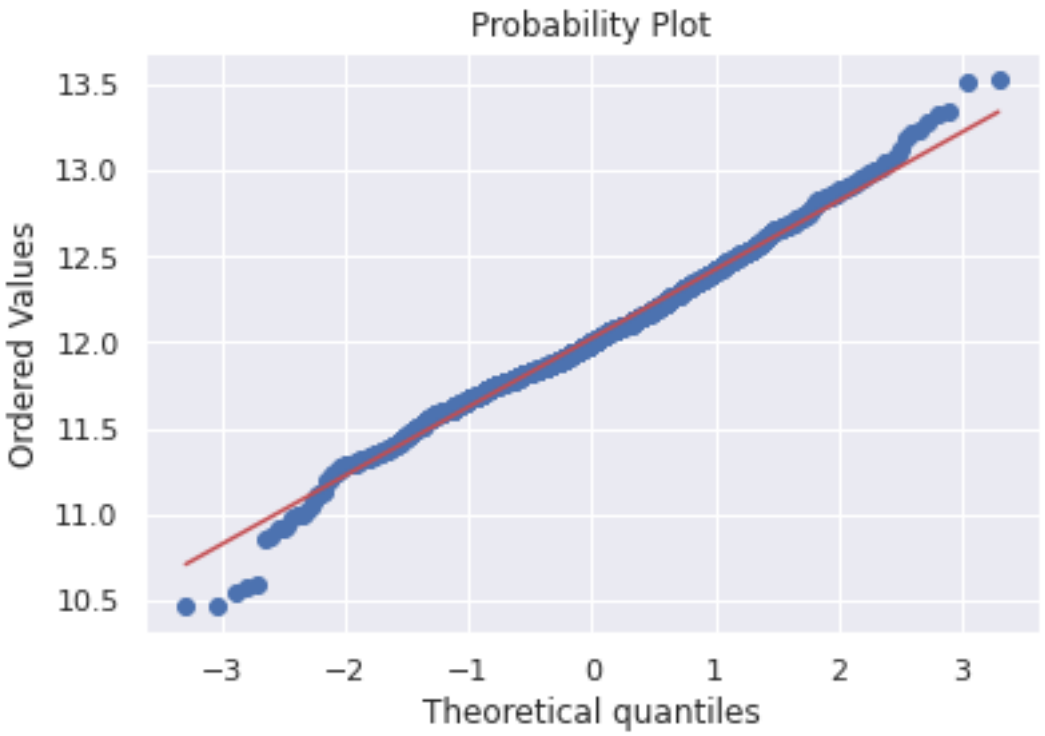


Similarly we will use same techniqe to normalize the skiwness of the other features.

1.6.2 3.6.2 Above Ground Living Area

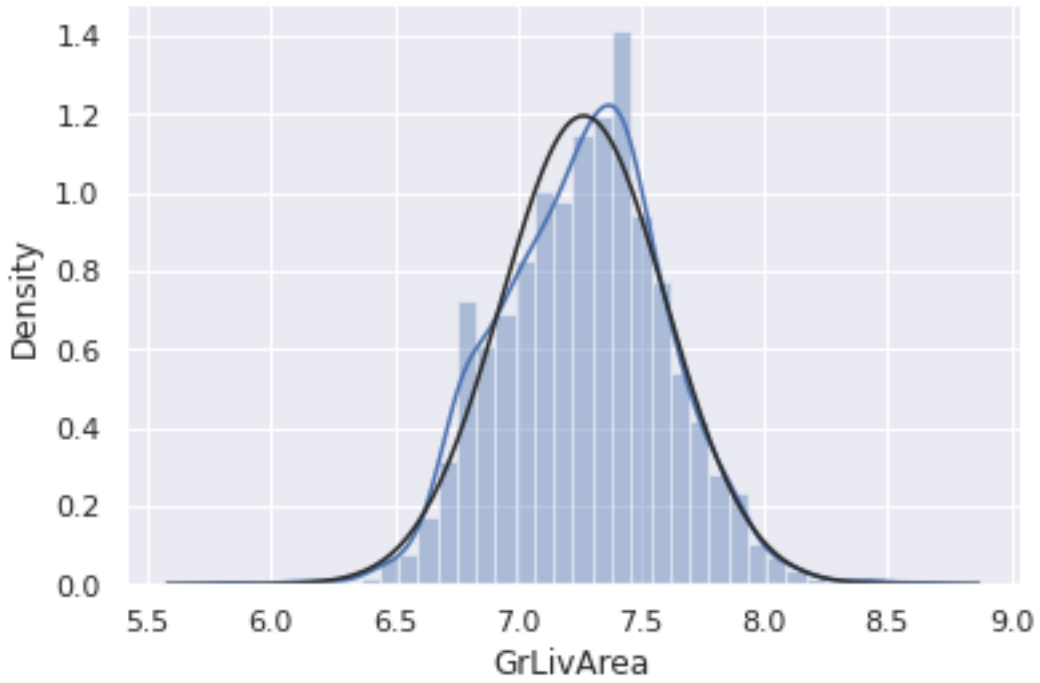
[19]:
```
# Above grade (ground) living area square feet
sns.distplot(df['GrLivArea'], fit=norm);
fig = plt.figure()
res = stats.probplot(df['SalePrice'], plot=plt)
```

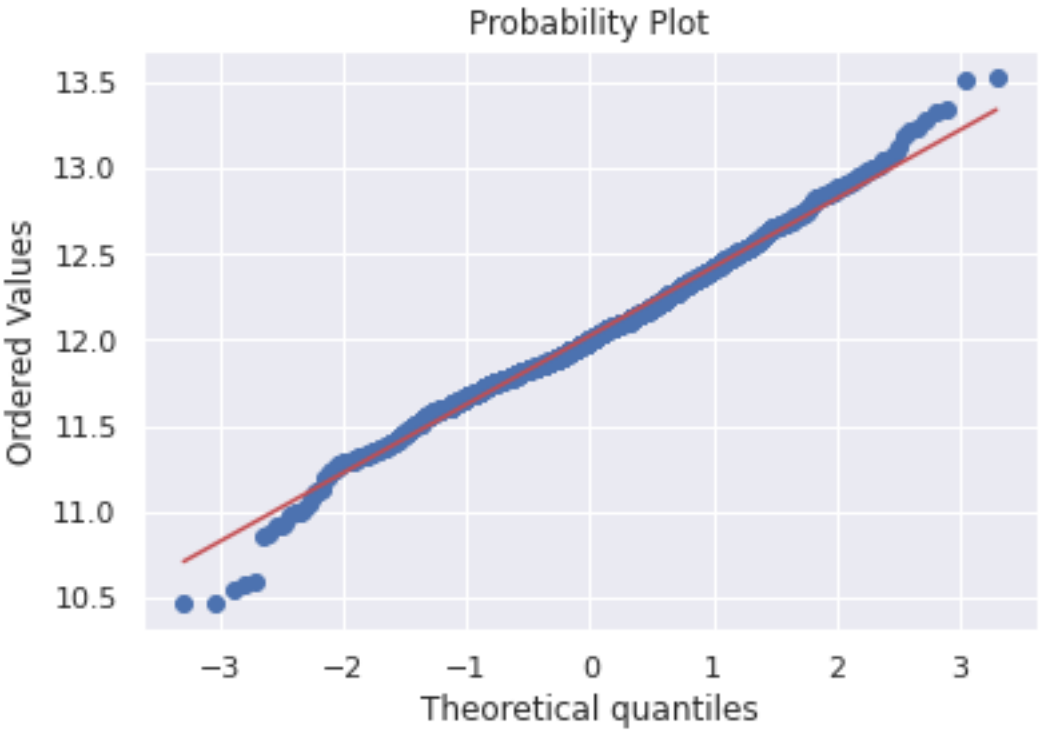




There is a positive skwness in the feature. We need to normalize the feature.

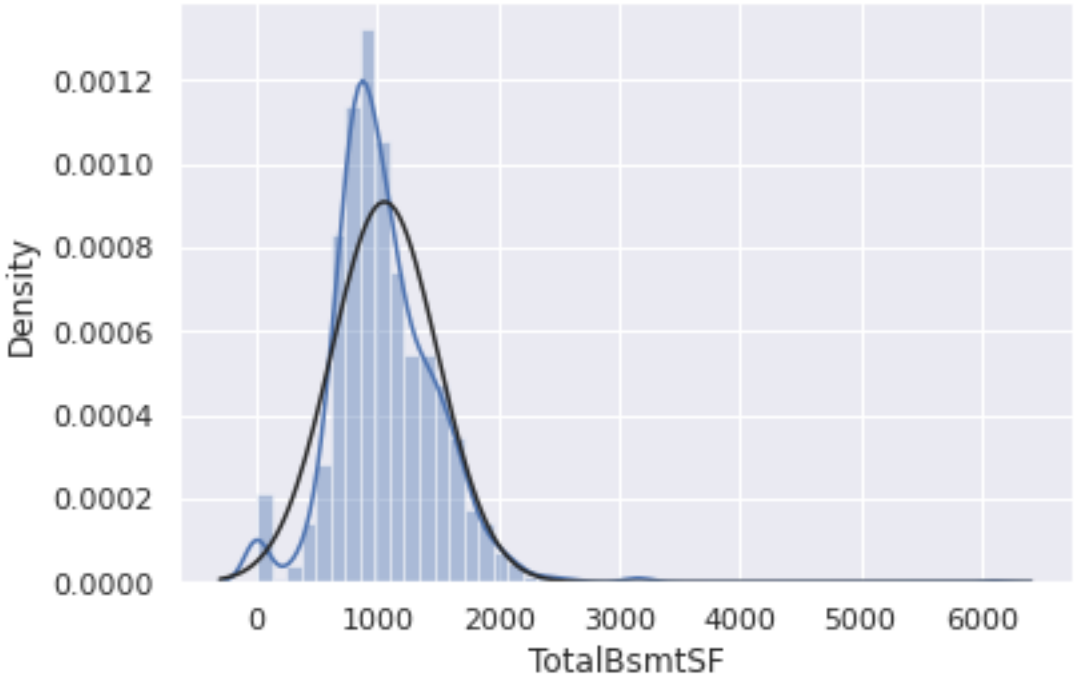
[20]:
```
# Normalizing Above grade (ground) living area square feet
df['GrLivArea'] = np.log(df['GrLivArea'])
sns.distplot(df['GrLivArea'], fit=norm);
fig = plt.figure()
res = stats.probplot(df['SalePrice'], plot=plt)
```

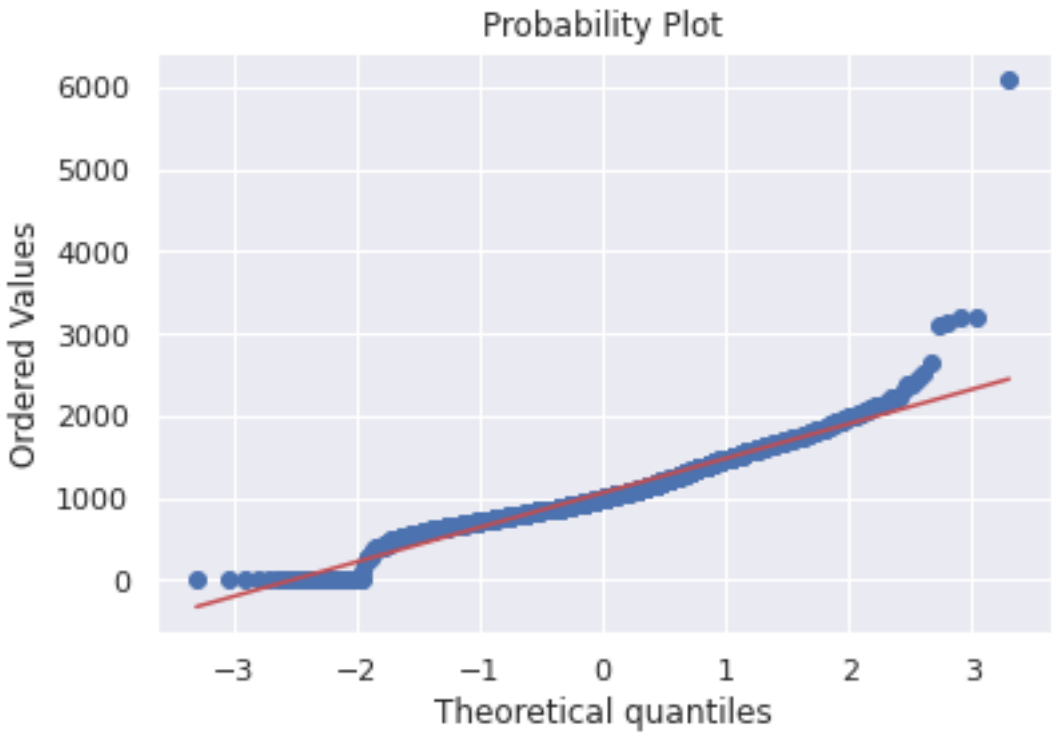

Density
GrLivArea
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
5.5
6.0
6.5
7.0
7.5
8.0
8.5
9.0


Probability Plot
Ordered Values
Theoretical quantiles
13.5
13.0
12.5
12.0
11.5
11.0
10.5
−3
−2
−1
0
1
2
3

1.6.3 3.6.3 Total basement area

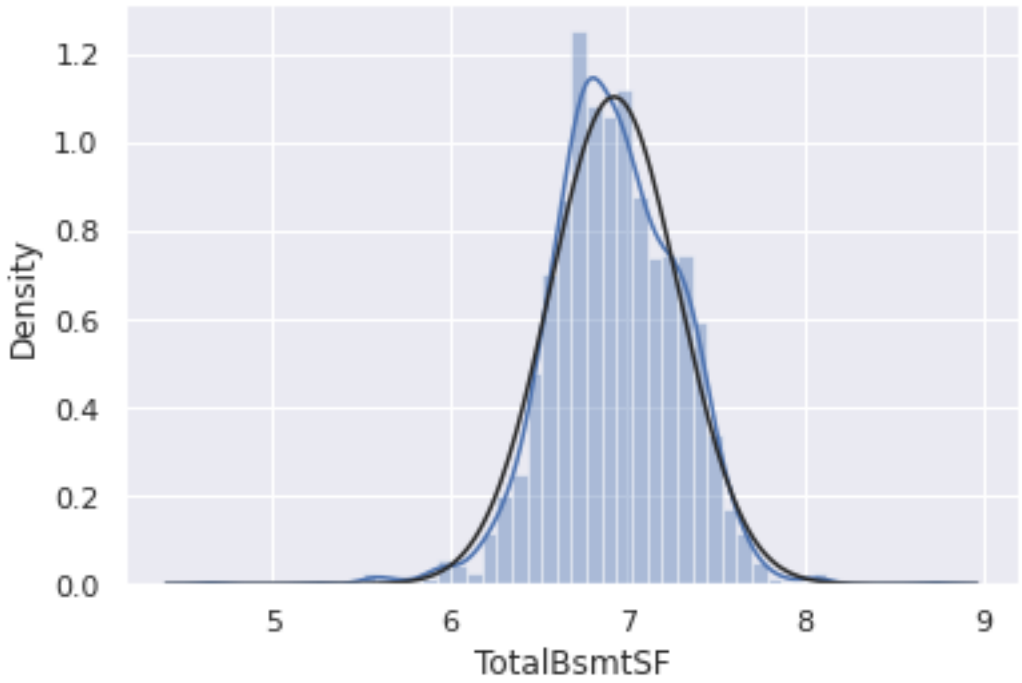
[21]:
```
sns.distplot(df['TotalBsmtSF'], fit=norm);
fig = plt.figure()
res = stats.probplot(df['TotalBsmtSF'], plot=plt)
```

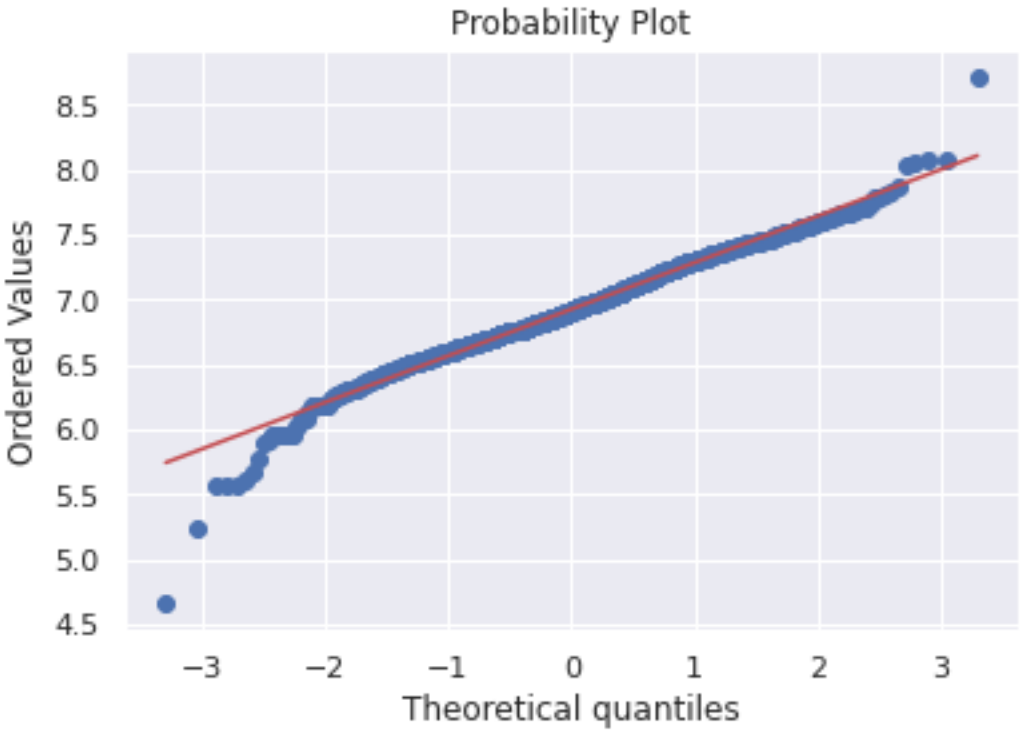




In TotalBsmtSF alot of datapoints have zero values, we have to log transform without affecting them.

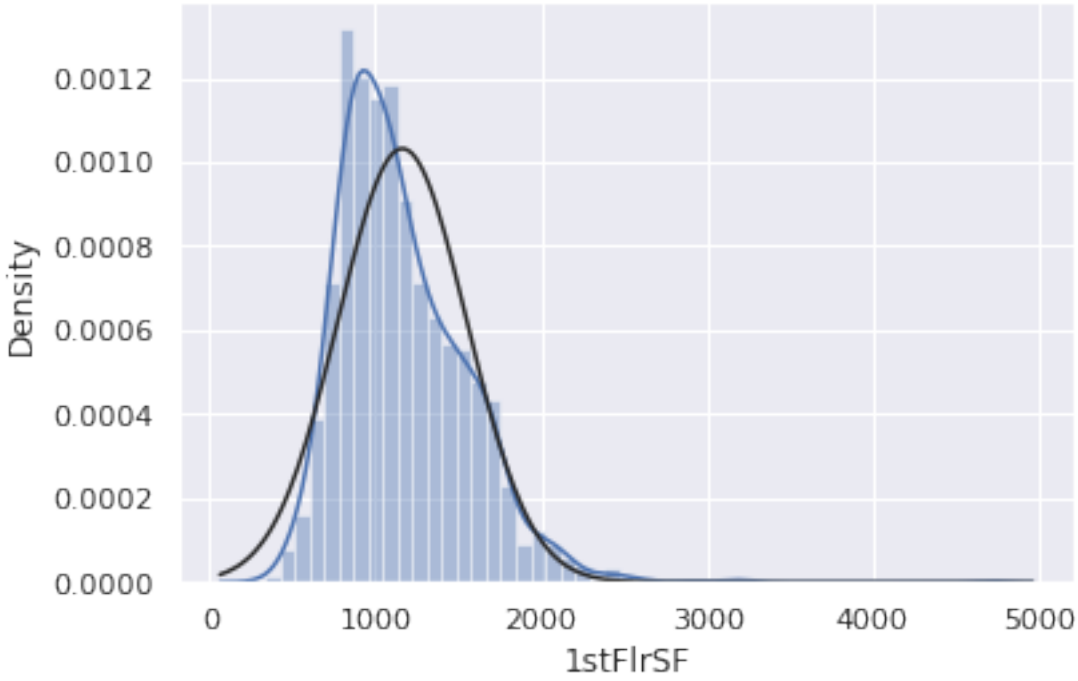
[22]:
```
df['TotalBsmtSF'] = df['TotalBsmtSF'].apply(lambda x: np.log(x) if x != 0 else␣
↪x)

sns.distplot(df[df['TotalBsmtSF']>0]['TotalBsmtSF'], fit=norm);
fig = plt.figure()
res = stats.probplot(df[df['TotalBsmtSF']>0]['TotalBsmtSF'], plot=plt)
```

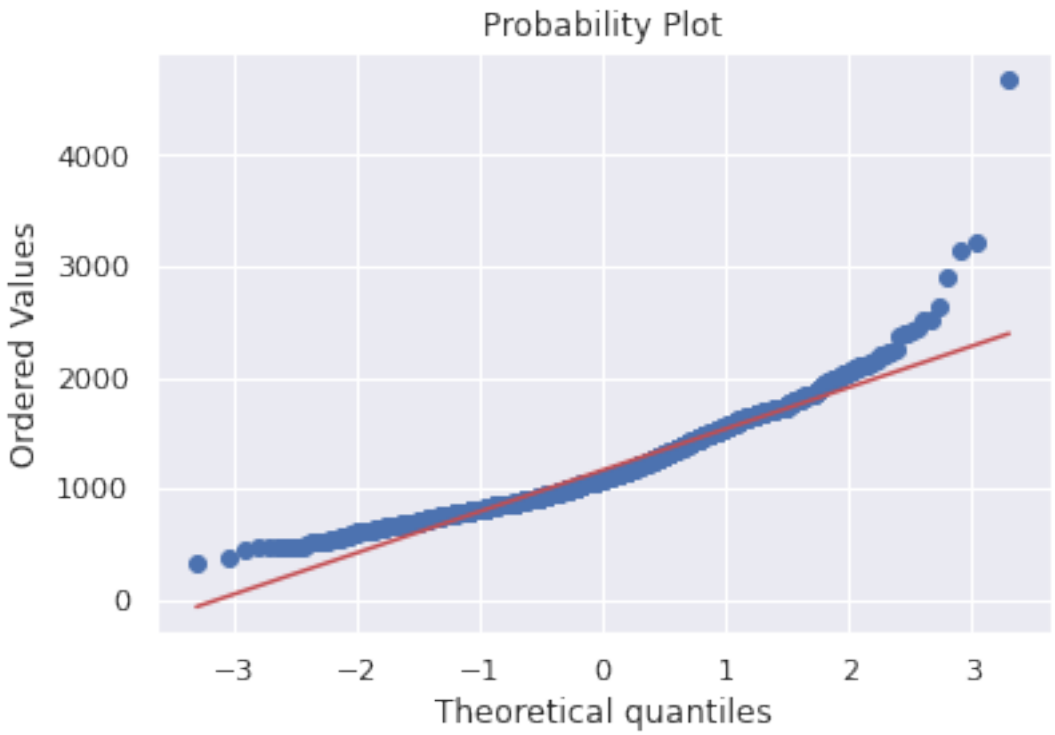

1.2
1.0
0.8
0.6
0.4
0.2
0.0
Density
5
6
7
8
9
TotalBsmtSF


Probability Plot
8.5
8.0
7.5
7.0
6.5
6.0
5.5
5.0
4.5
Ordered Values
−3
−2
−1
0
1
2
3
Theoretical quantiles

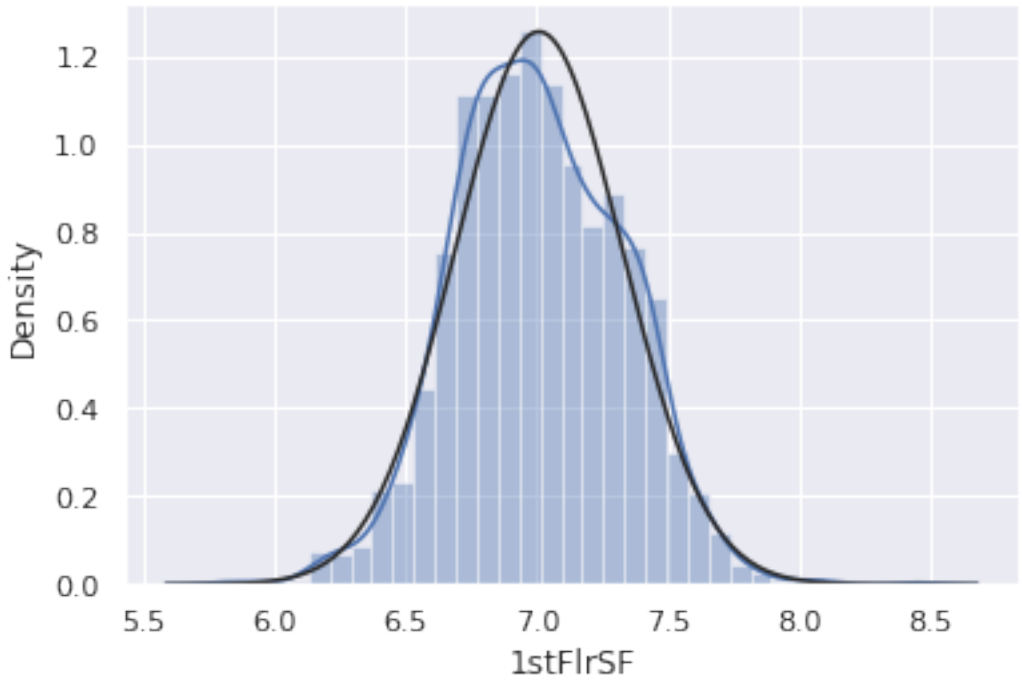
1.6.4 3.6.5 First Floor Area

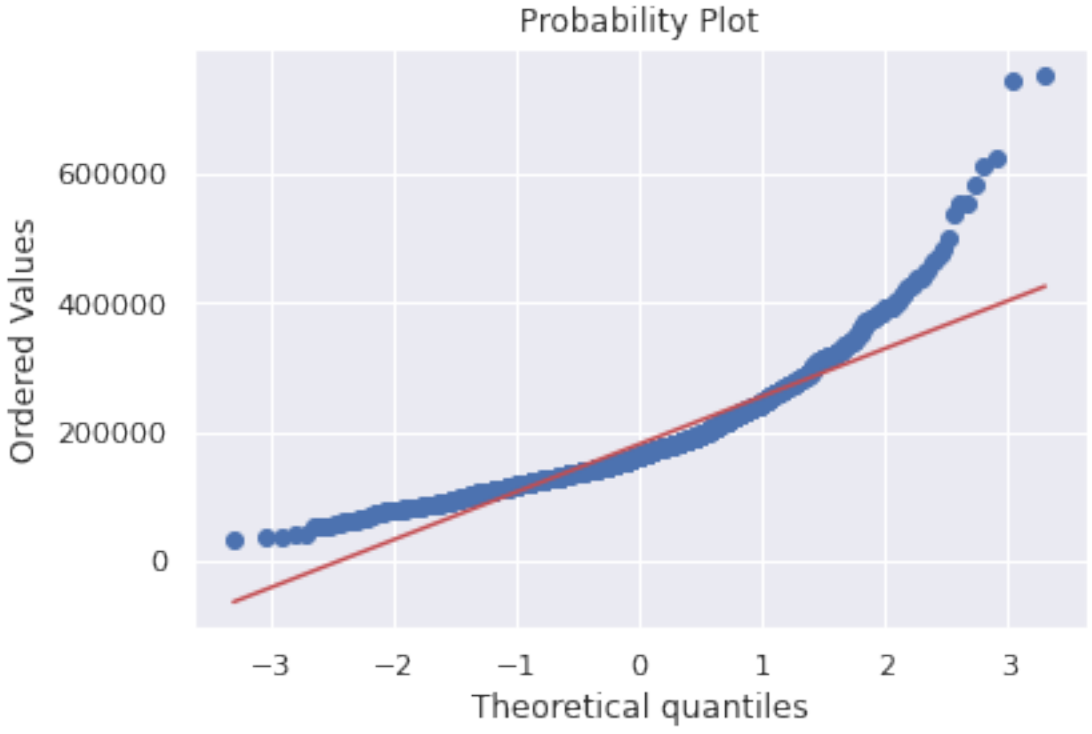
[39]:
```
sns.distplot(df['1stFlrSF'], fit=norm);
fig = plt.figure()
res = stats.probplot(df['1stFlrSF'], plot=plt)
```





[40]:
```
df['1stFlrSF'] = np.log(df['1stFlrSF'])
sns.distplot(df['1stFlrSF'], fit=norm);
fig = plt.figure()
res = stats.probplot(df['SalePrice'], plot=plt)
```


1.2
1.0
0.8
0.6
0.4
0.2
0.0
Density
5.5
6.0
6.5
7.0
7.5
8.0
8.5
1stFlrSF


Probability Plot
600000
400000
200000
0
Ordered Values
−3
−2
−1
0
1
2
3
Theoretical quantiles

[ ]:
```
df.to_csv('final.csv')
```