

Feature Engineering

September 3, 2022

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

```
[4]: train=pd.read_csv("PEP1.csv")
```

```
[5]: train.describe(include='all')
```

```
[5]:
```

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	\
count	1460.000000	1460.000000	1460	1201.000000	1460.000000	1460	
unique	NaN	NaN	5	NaN	NaN	2	
top	NaN	NaN	RL	NaN	NaN	Pave	
freq	NaN	NaN	1151	NaN	NaN	1454	
mean	730.500000	56.897260	NaN	70.049958	10516.828082	NaN	
std	421.610009	42.300571	NaN	24.284752	9981.264932	NaN	
min	1.000000	20.000000	NaN	21.000000	1300.000000	NaN	
25%	365.750000	20.000000	NaN	59.000000	7553.500000	NaN	
50%	730.500000	50.000000	NaN	69.000000	9478.500000	NaN	
75%	1095.250000	70.000000	NaN	80.000000	11601.500000	NaN	
max	1460.000000	190.000000	NaN	313.000000	215245.000000	NaN	

	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	\
count	91	1460	1460	1460	...	1460.000000	7	281	
unique	2	4	4	2	...	NaN	3	4	
top	Grvl	Reg	Lvl	AllPub	...	NaN	Gd	MnPrv	
freq	50	925	1311	1459	...	NaN	3	157	
mean	NaN	NaN	NaN	NaN	...	2.758904	NaN	NaN	
std	NaN	NaN	NaN	NaN	...	40.177307	NaN	NaN	
min	NaN	NaN	NaN	NaN	...	0.000000	NaN	NaN	
25%	NaN	NaN	NaN	NaN	...	0.000000	NaN	NaN	
50%	NaN	NaN	NaN	NaN	...	0.000000	NaN	NaN	
75%	NaN	NaN	NaN	NaN	...	0.000000	NaN	NaN	
max	NaN	NaN	NaN	NaN	...	738.000000	NaN	NaN	

	MiscFeature	MiscVal	MoSold	YrSold	SaleType	\
count	54	1460.000000	1460.000000	1460.000000	1460	

unique	4	NaN	NaN	NaN	9
top	Shed	NaN	NaN	NaN	WD
freq	49	NaN	NaN	NaN	1267
mean	NaN	43.489041	6.321918	2007.815753	NaN
std	NaN	496.123024	2.703626	1.328095	NaN
min	NaN	0.000000	1.000000	2006.000000	NaN
25%	NaN	0.000000	5.000000	2007.000000	NaN
50%	NaN	0.000000	6.000000	2008.000000	NaN
75%	NaN	0.000000	8.000000	2009.000000	NaN
max	NaN	15500.000000	12.000000	2010.000000	NaN

	SaleCondition	SalePrice
count	1460	1460.000000
unique	6	NaN
top	Normal	NaN
freq	1198	NaN
mean	NaN	180921.195890
std	NaN	79442.502883
min	NaN	34900.000000
25%	NaN	129975.000000
50%	NaN	163000.000000
75%	NaN	214000.000000
max	NaN	755000.000000

[11 rows x 81 columns]

```
[6]: train.shape
```

```
[6]: (1460, 81)
```

```
[7]: train.isnull().sum().sort_values(ascending= False).head(20)
```

```
[7]: PoolQC          1453
MiscFeature        1406
Alley              1369
Fence              1179
FireplaceQu        690
LotFrontage        259
GarageCond          81
GarageType          81
GarageYrBlt         81
GarageFinish        81
GarageQual          81
BsmtExposure        38
BsmtFinType2         38
BsmtFinType1         37
BsmtCond             37
```

```

BsmtQual      37
MasVnrArea    8
MasVnrType    8
Electrical    1
Utilities     0
dtype: int64

```

```
[9]: train.drop(['PoolQC', 'MiscFeature', 'Alley', 'Fence', 'LotFrontage'],axis=1)
```

```

[9]:      Id  MSSubClass MSZoning  LotArea  Street  LotShape  LandContour  \
0        1         60      RL      8450   Pave      Reg      Lvl
1        2         20      RL     9600   Pave      Reg      Lvl
2        3         60      RL    11250   Pave      IR1      Lvl
3        4         70      RL     9550   Pave      IR1      Lvl
4        5         60      RL    14260   Pave      IR1      Lvl
...    ...      ...      ...      ...      ...      ...      ...
1455  1456         60      RL     7917   Pave      Reg      Lvl
1456  1457         20      RL    13175   Pave      Reg      Lvl
1457  1458         70      RL     9042   Pave      Reg      Lvl
1458  1459         20      RL     9717   Pave      Reg      Lvl
1459  1460         20      RL     9937   Pave      Reg      Lvl

      Utilities LotConfig LandSlope  ... EnclosedPorch 3SsnPorch ScreenPorch  \
0      AllPub   Inside    Gtl  ...           0           0           0
1      AllPub    FR2     Gtl  ...           0           0           0
2      AllPub   Inside    Gtl  ...           0           0           0
3      AllPub  Corner    Gtl  ...          272           0           0
4      AllPub    FR2     Gtl  ...           0           0           0
...    ...      ...      ...      ...      ...      ...      ...
1455  AllPub   Inside    Gtl  ...           0           0           0
1456  AllPub   Inside    Gtl  ...           0           0           0
1457  AllPub   Inside    Gtl  ...           0           0           0
1458  AllPub   Inside    Gtl  ...          112           0           0
1459  AllPub   Inside    Gtl  ...           0           0           0

      PoolArea MiscVal  MoSold  YrSold  SaleType  SaleCondition  SalePrice
0           0         0        2   2008        WD        Normal    208500
1           0         0        5   2007        WD        Normal    181500
2           0         0        9   2008        WD        Normal    223500
3           0         0        2   2006        WD      Abnorml    140000
4           0         0       12   2008        WD        Normal    250000
...    ...      ...      ...      ...      ...      ...      ...
1455         0         0        8   2007        WD        Normal    175000
1456         0         0        2   2010        WD        Normal    210000
1457         0      2500        5   2010        WD        Normal    266500
1458         0         0        4   2010        WD        Normal    142125
1459         0         0        6   2008        WD        Normal    147500

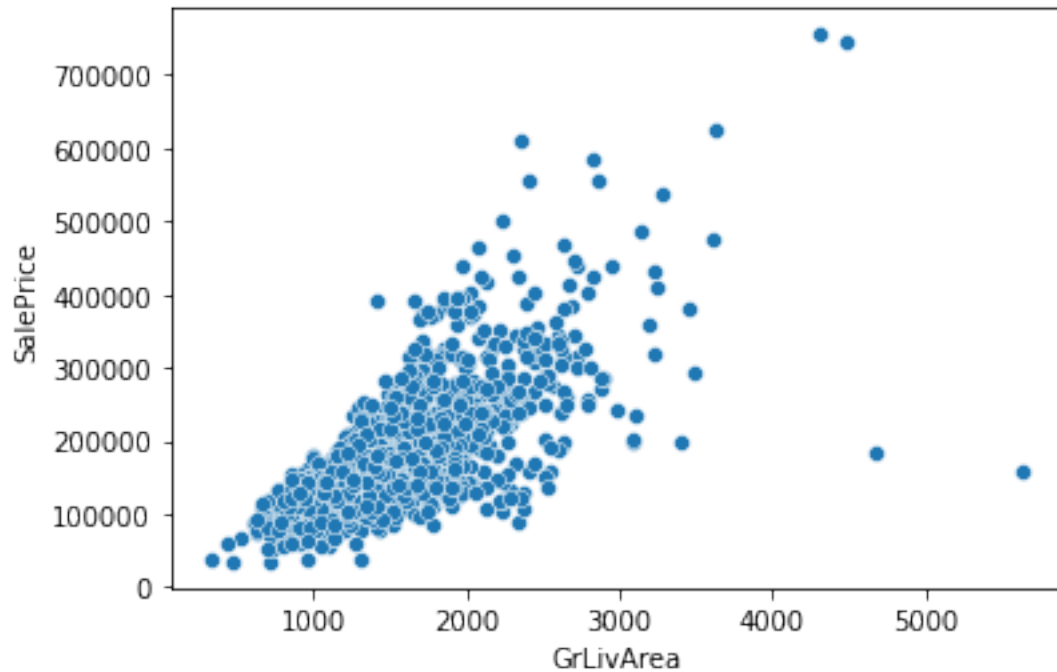
```

[1460 rows x 76 columns]

```
[10]: print(train.shape)
sns.scatterplot(y="SalePrice", x="GrLivArea", data=train)
```

(1460, 81)

```
[10]: <AxesSubplot:xlabel='GrLivArea', ylabel='SalePrice'>
```

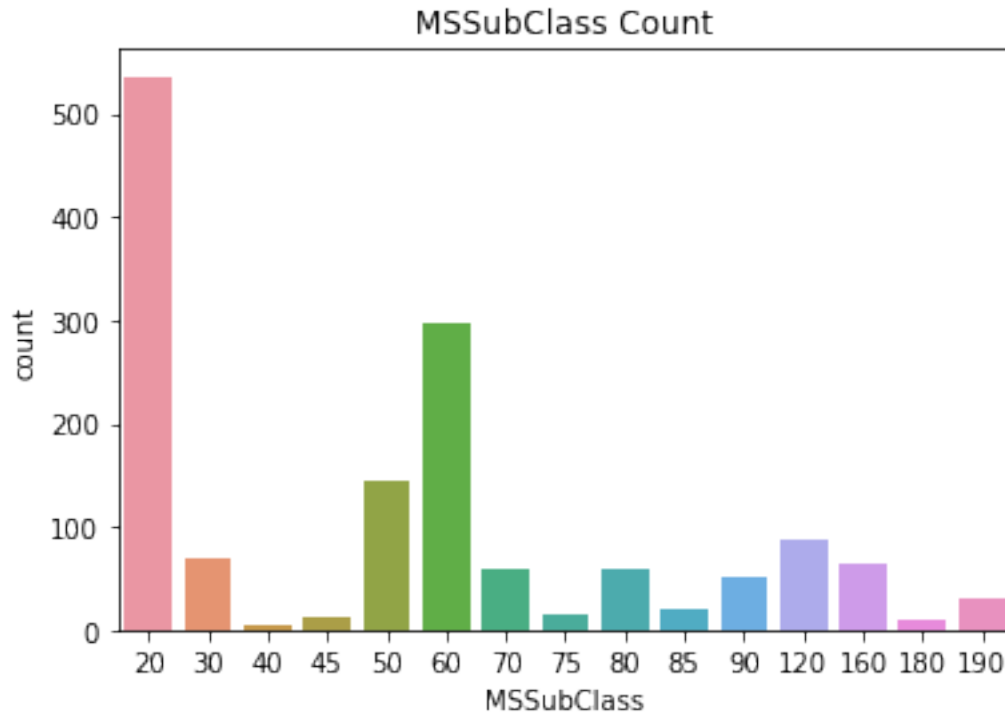


```
[11]: train.drop(train[(train['GrLivArea']>4000) & (train['SalePrice']<200000)].
        ↳index, inplace=True)
print('Rows and columns after dropping outliers :', train.shape )
```

Rows and columns after dropping outliers : (1458, 81)

```
[12]: sns.countplot(x="MSSubClass",data=train)
plt.title('MSSubClass Count')
```

```
[12]: Text(0.5, 1.0, 'MSSubClass Count')
```



```
[14]: cols = ['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']

features=[]
obj_col=[]
for col in cols:
    if (train[col].dtype) == object :
        obj_col.append(col)
obj_col
```



```

2904             if self.columns.nlevels > 1:
2905                 return self._getitem_multilevel(key)
-> 2906             indexer = self.columns.get_loc(key)
2907             if is_integer(indexer):
2908                 indexer = [indexer]

/usr/local/lib/python3.7/site-packages/pandas/core/indexes/base.py in
-> get_loc(self, key, method, tolerance)
2898             return self._engine.get_loc(casted_key)
2899             except KeyError as err:
-> 2900                 raise KeyError(key) from err
2901
2902             if tolerance is not None:

```

KeyError: 'KitchenAbvGr'

```

[15]: def plotting(col):
        sns.countplot(x=col,data=train)
        plt.title('CountPlot for ' + col)
        print('No. of missing values are : ')
        print(train[col].isnull().sum())

```

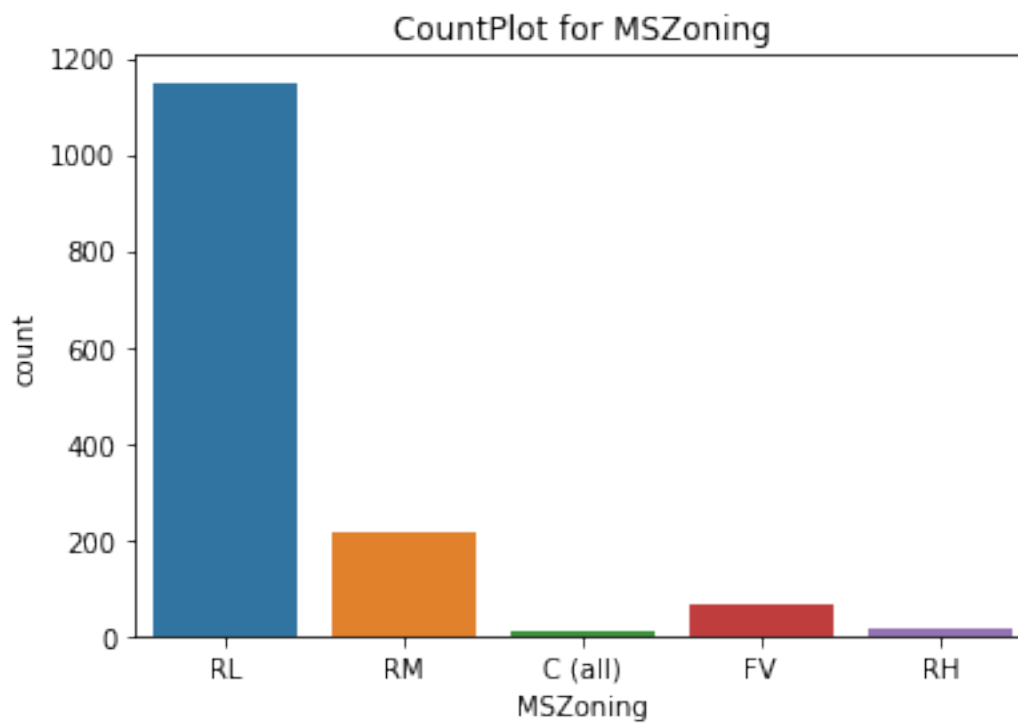
```

[16]: plotting('MSZoning')

```

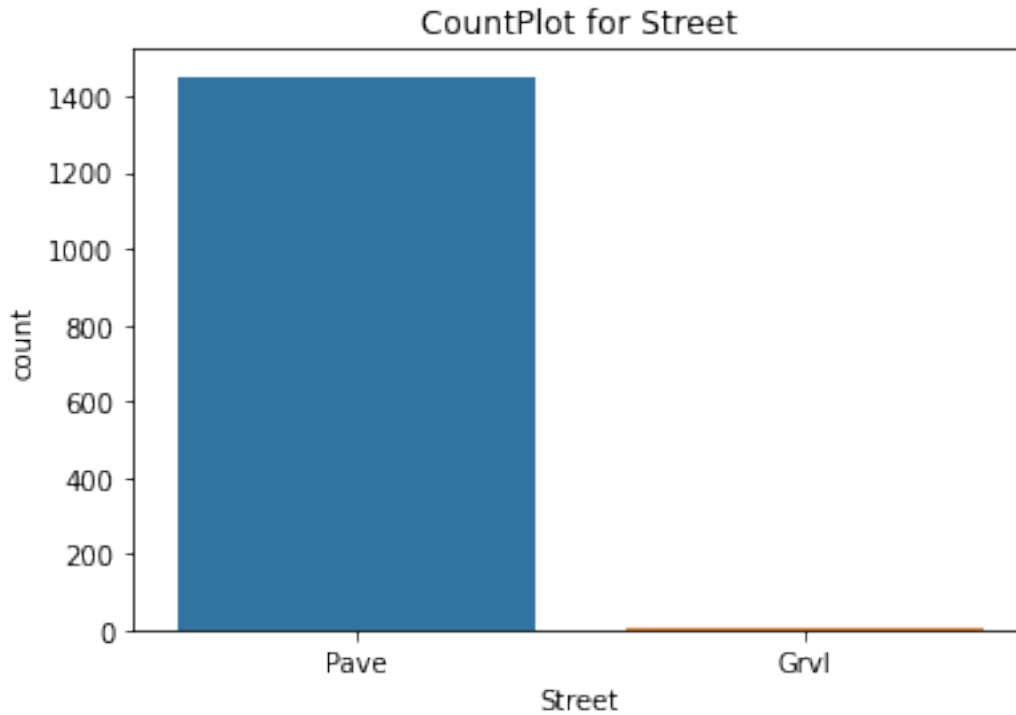
No. of missing values are :

0



```
[17]: plotting('Street')
```

No. of missing values are :
0



```
[18]: plotting('LotShape')
features.append('LotShape')
train.LotShape=train.LotShape.replace({'Reg':1, 'IR1':2, 'IR2':3, 'IR3':4})
test.LotShape=test.LotShape.replace({'Reg':1, 'IR1':2, 'IR2':3, 'IR3':4})
```

No. of missing values are :
0

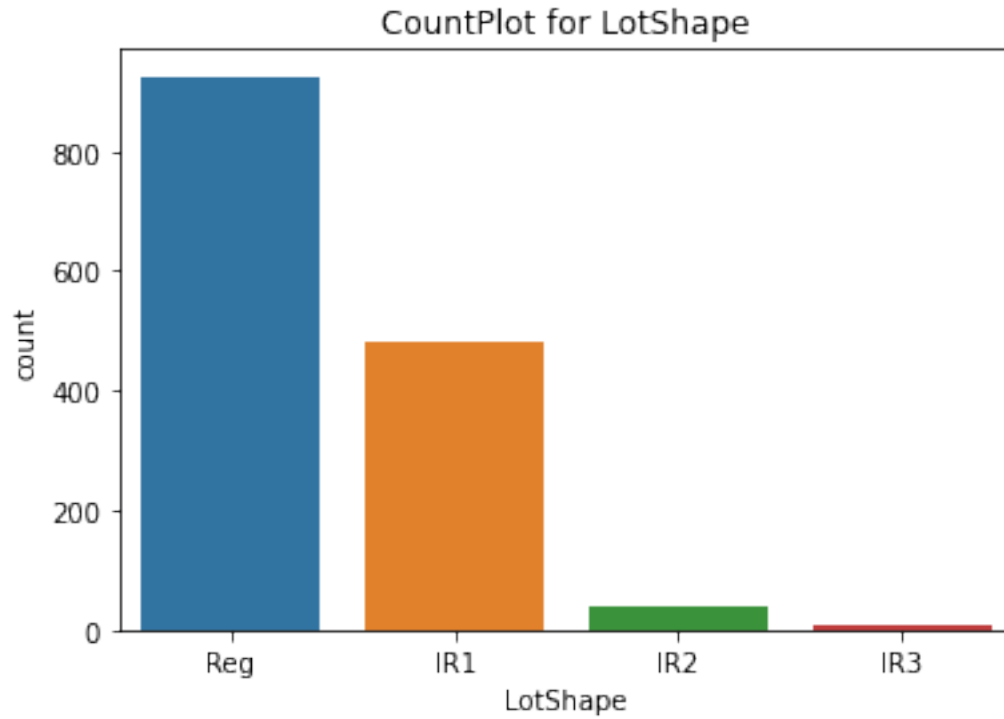
```

↳
-----
NameError                                Traceback (most recent call↳
↳last)

<ipython-input-18-7f7083f5dddc> in <module>
      2 features.append('LotShape')
      3 train.LotShape=train.LotShape.replace({'Reg':1, 'IR1':2, 'IR2':3, 'IR3':
↳4})
----> 4 test.LotShape=test.LotShape.replace({'Reg':1, 'IR1':2, 'IR2':3, 'IR3':
↳4})

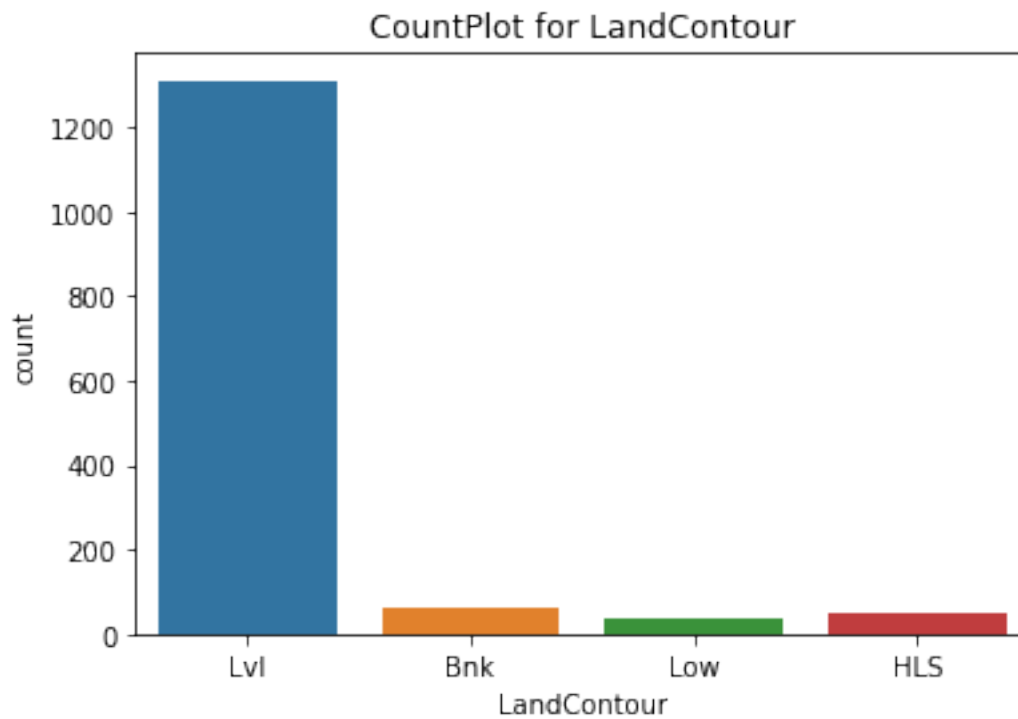
NameError: name 'test' is not defined

```



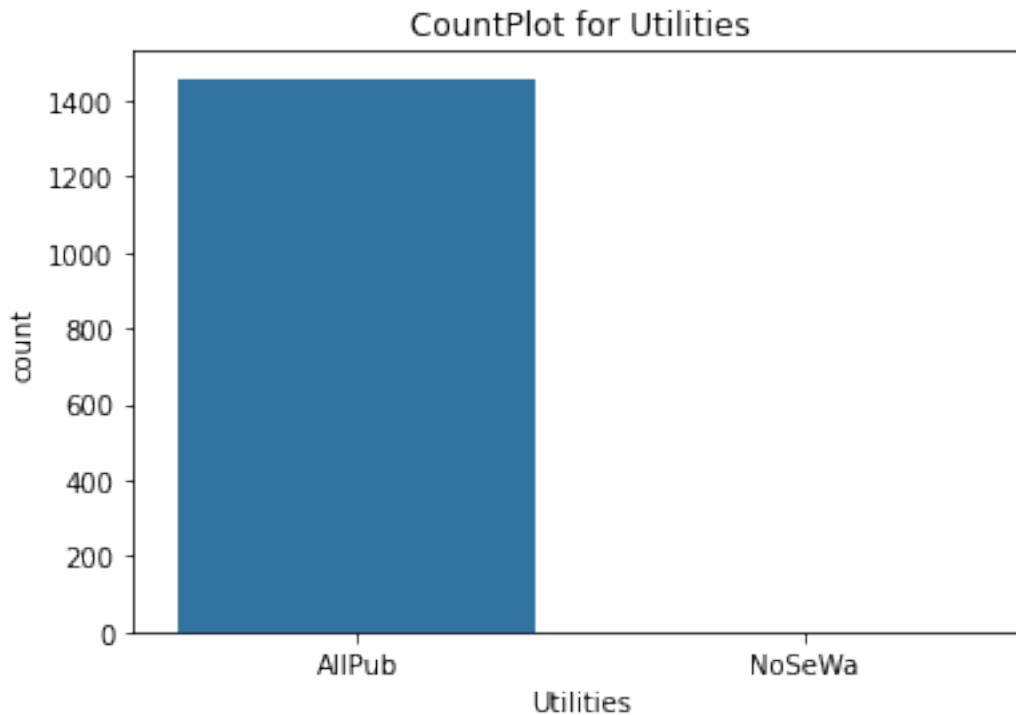
```
[19]: plotting('LandContour')
```

No. of missing values are :
0



```
[20]: plotting('Utilities')
```

No. of missing values are :
0



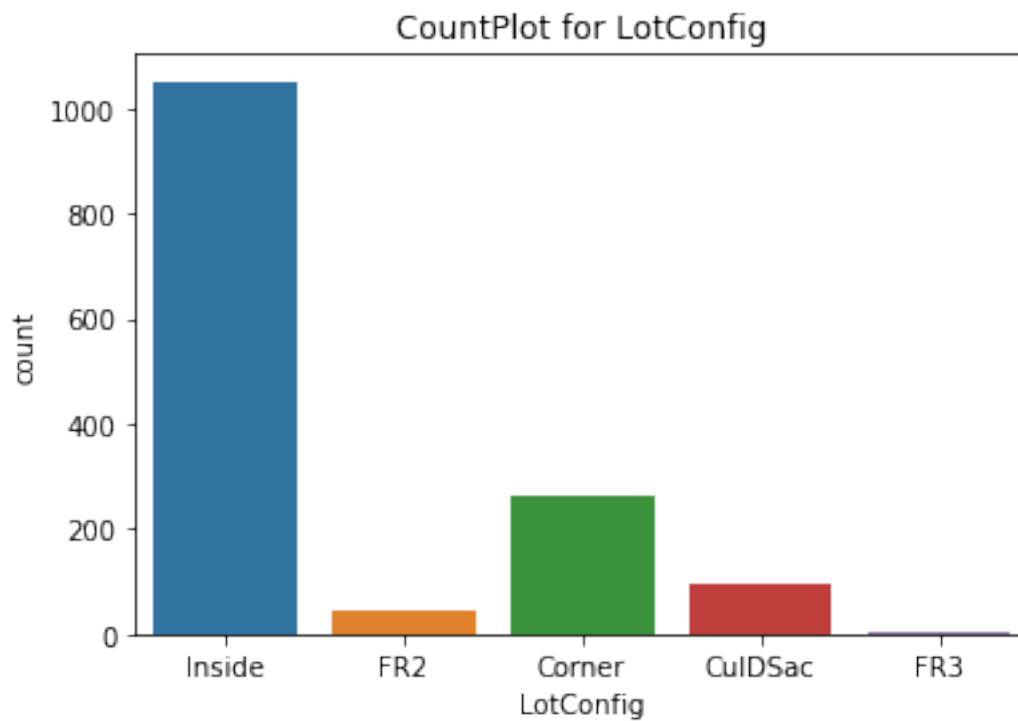
```
[21]: plotting('LotConfig')
      features.append('LotConfig')

      train.LotConfig=train.LotConfig.replace({'Inside':1,'Corner':2,'CulDSac':
      ↳3,'FR2':4,'FR3':5})
      test.LotConfig=test.LotConfig.replace({'Inside':1,'Corner':2,'CulDSac':3,'FR2':
      ↳4,'FR3':5})
```

No. of missing values are :
0

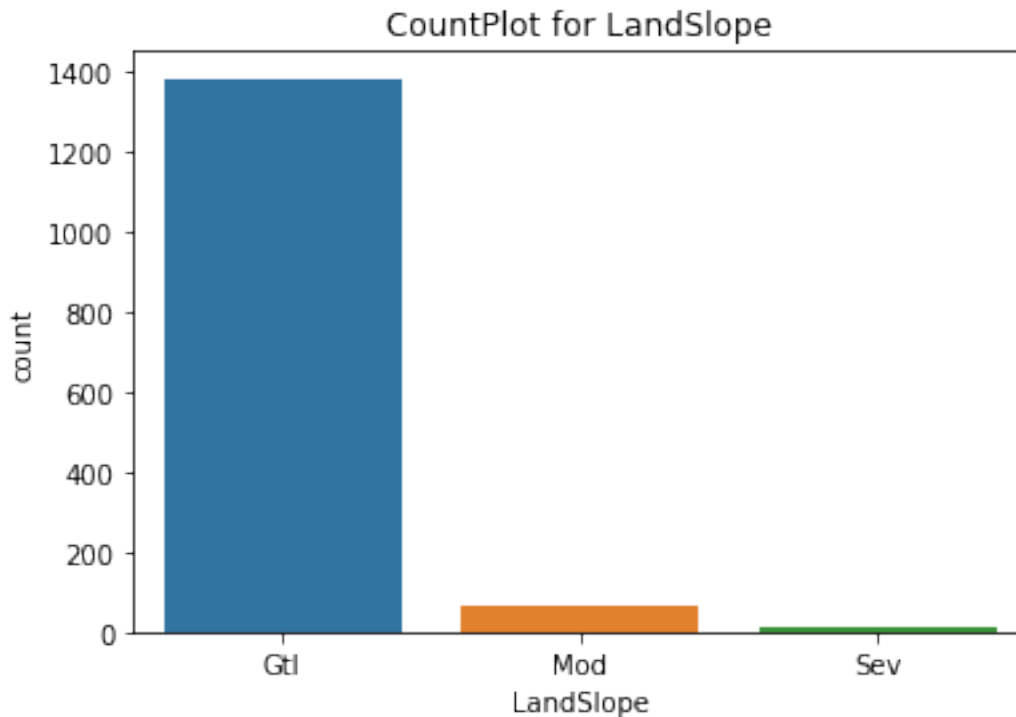
```
↳
↳ -----
↳
↳ NameError                                Traceback (most recent call↳
↳ last)
↳
↳ <ipython-input-21-d7bd811a34f4> in <module>
↳     3
↳     4 train.LotConfig=train.LotConfig.replace({'Inside':1,'Corner':
↳ ↳2,'CulDSac':3,'FR2':4,'FR3':5})
↳ ----> 5 test.LotConfig=test.LotConfig.replace({'Inside':1,'Corner':
↳ ↳2,'CulDSac':3,'FR2':4,'FR3':5})
```

NameError: name 'test' is not defined



```
[22]: plotting('LandSlope')
```

No. of missing values are :
0



```
[23]: train_neighbor=train.copy()
print(train_neighbor[['Neighborhood','SalePrice']].
      ↳groupby(['Neighborhood'],as_index=True).mean())
plotting('Neighborhood')
#This looks promising as Price varies from area to area and it can be seen
↳through the mean of the prices of selected Neighborhood.
features.append('Neighborhood')
train.Neighborhood=train.Neighborhood.replace({'MeadowV':1,
                                              'IDOTRR':2, 'BrDale':2,
                                              'OldTown':3, 'Edwards':3,
                                              ↳'BrkSide':3,
                                              'Sawyer':4, 'Blueste':4, 'SWISU':
                                              ↳4, 'NAmes':4,
                                              'NPkVill':5, 'Mitchel':5,
                                              'SawyerW':6, 'Gilbert':6,
                                              ↳'NWAmes':6,
                                              'Blmngtn':7, 'CollgCr':7,
                                              ↳'ClearCr':7, 'Crawfor':7,
                                              'Veenker':8, 'Somerst':8,
                                              ↳'Timber':8,
                                              'StoneBr':9,
                                              'NoRidge':10, 'NridgHt':10})
test.Neighborhood=test.Neighborhood.replace({'MeadowV':1,
```

```

↪ 'BrkSide':3,
↪4, 'NAmes':4,
↪ 'NWAmes':6,
↪ 'ClearCr':7, 'Crawfor':7,
↪ 'Timber':8,
'IDOTRR':2, 'BrDale':2,
'OldTown':3, 'Edwards':3,␣
'Sawyer':4, 'Blueste':4, 'SWISU':
'NPkVill':5, 'Mitchel':5,
'SawyerW':6, 'Gilbert':6,␣
'Blmngtn':7, 'CollgCr':7,␣
'Veenker':8, 'Somerst':8,␣
'StoneBr':9,
'NoRidge':10, 'NridgHt':10})

```

Neighborhood	SalePrice
Blmngtn	194870.882353
Blueste	137500.000000
BrDale	104493.750000
BrkSide	124834.051724
ClearCr	212565.428571
CollgCr	197965.773333
Crawfor	210624.725490
Edwards	127318.571429
Gilbert	192854.506329
IDOTRR	100123.783784
MeadowV	98576.470588
Mitchel	156270.122449
NPkVill	142694.444444
NWAmes	189050.068493
NoRidge	335295.317073
NridgHt	316270.623377
OldTown	128225.300885
SWISU	142591.360000
Sawyer	136793.135135
SawyerW	186555.796610
Somerst	225379.837209
StoneBr	310499.000000
Timber	242247.447368
Veenker	238772.727273
mes	145847.080000
No. of missing values are :	
0	

↪ -----

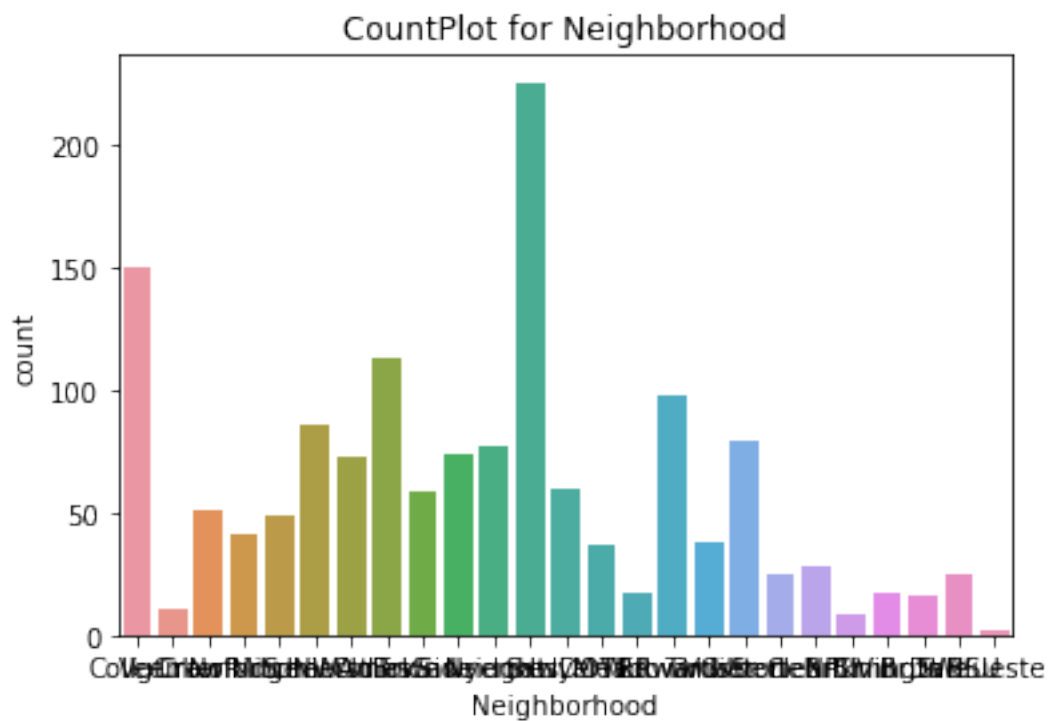
```

NameError                                Traceback (most recent call
↳ last)

<ipython-input-23-ba9417805f97> in <module>
    14                                     'StoneBr':9,
    15                                     'NoRidge':10,
↳ 'NridgHt':10})
    ---> 16 test.Neighborhood=test.Neighborhood.replace({'MeadowV':1,
    17                                                         'IDOTRR':2, 'BrDale':
↳ 2,
    18                                                         'OldTown':3,
↳ 'Edwards':3, 'BrkSide':3,

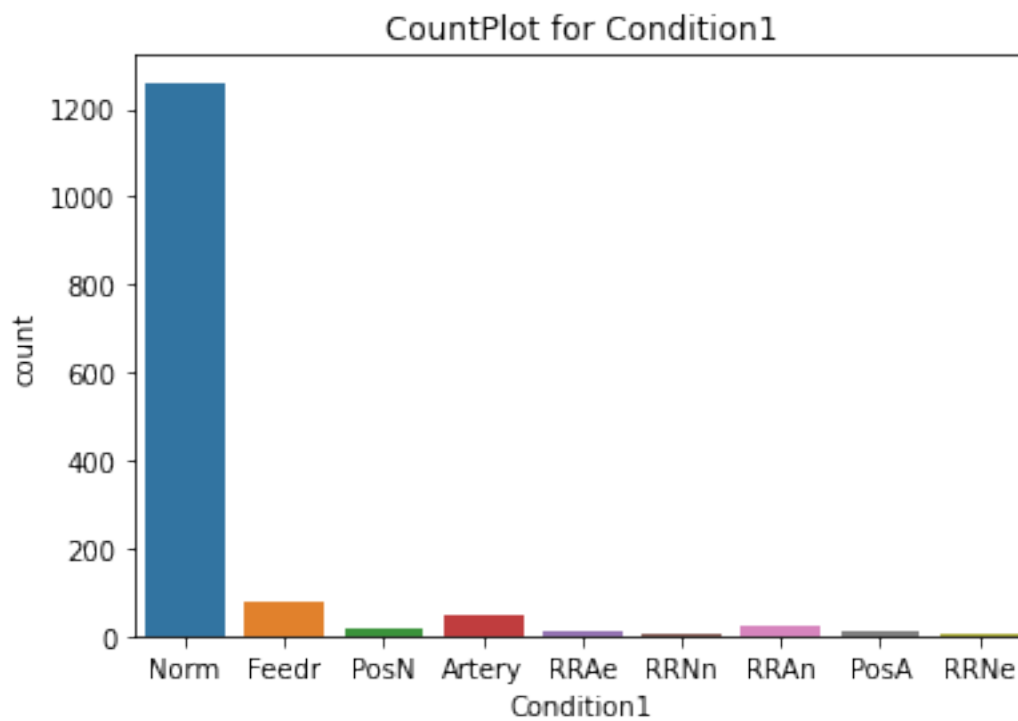
```

NameError: name 'test' is not defined



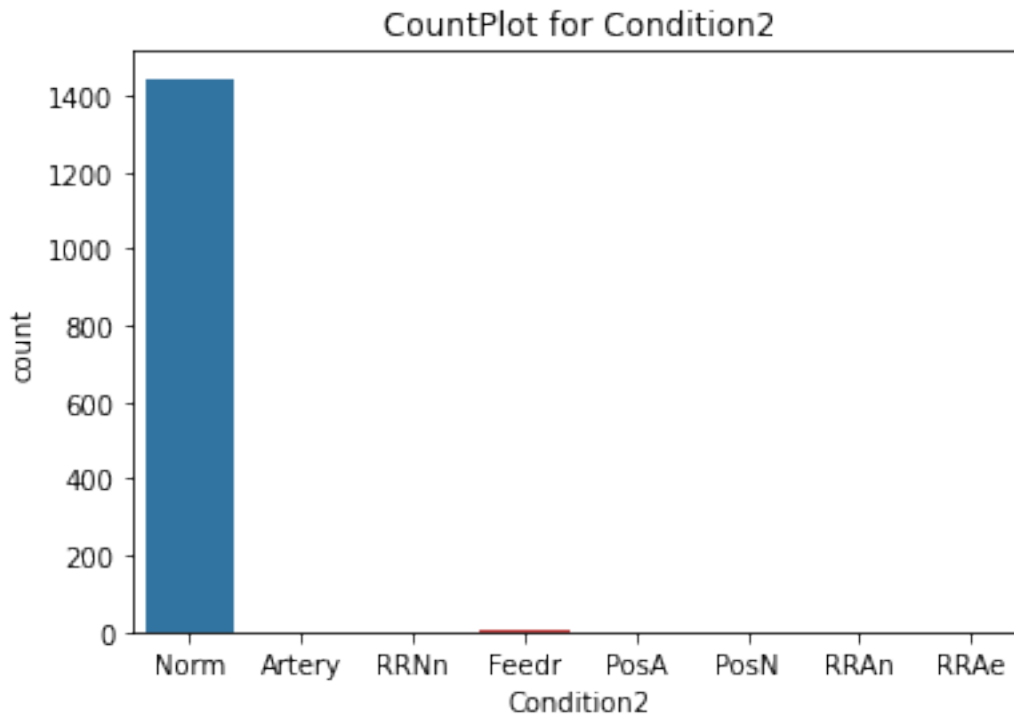
```
[24]: plotting('Condition1')
```

No. of missing values are :
0



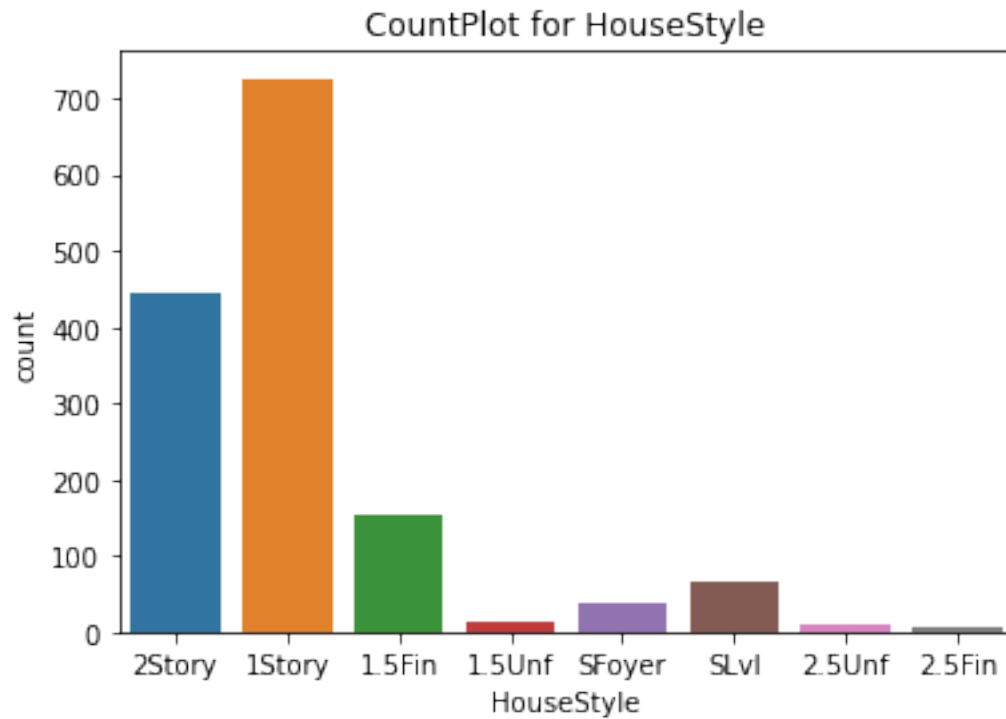
```
[25]: plotting('Condition2')
```

No. of missing values are :
0



```
[26]: plotting('HouseStyle')
plt.show()
sns.lineplot(y='SalePrice',x='HouseStyle',data=train)
#Clearly it can be seen from the plots that HouseStyle has big effect on
↪Saleprice.
features.append('HouseStyle')
#Mapping values
train.HouseStyle=train.HouseStyle.replace({'1.5Unf':1,
                                             '1.5Fin':2, '2.5Unf':2, 'SFoyer':2,
                                             '1Story':3, 'SLvl':3,
                                             '2Story':4, '2.5Fin':4})
test.HouseStyle=test.HouseStyle.replace({'1.5Unf':1,
                                           '1.5Fin':2, '2.5Unf':2, 'SFoyer':2,
                                           '1Story':3, 'SLvl':3,
                                           '2Story':4, '2.5Fin':4})
```

No. of missing values are :
0



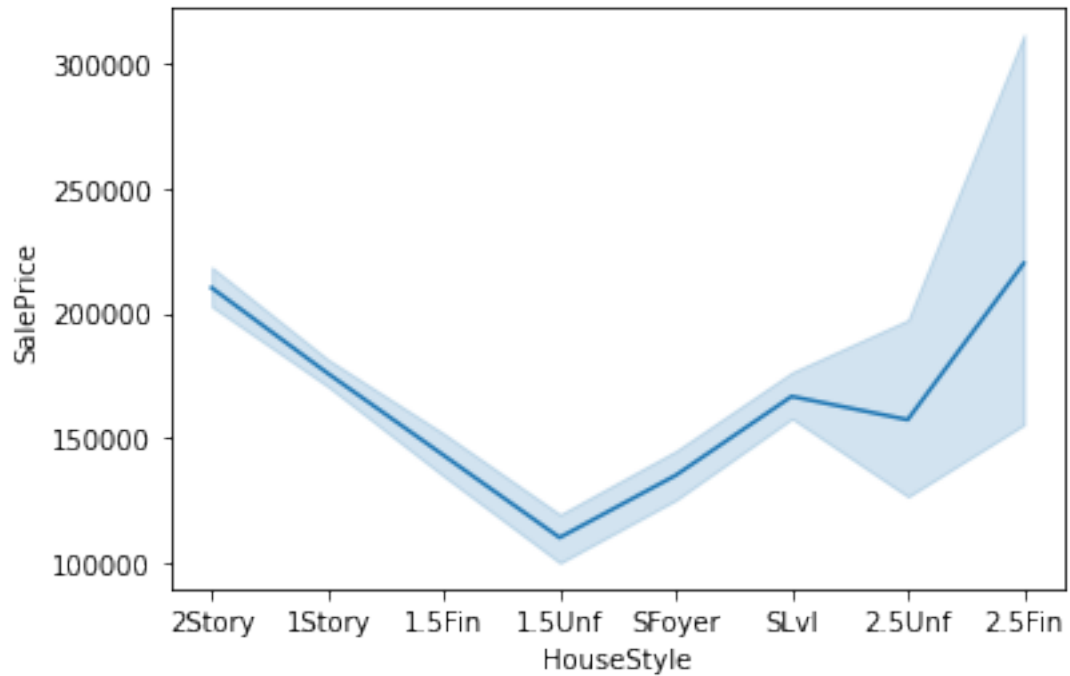
```

↳
-----
NameError                                Traceback (most recent call↳
↳last)

<ipython-input-26-b7969968da20> in <module>
     9                                     '1Story':3, 'SLvl':3,
    10                                     '2Story':4, '2.5Fin':4})
--> 11 test.HouseStyle=test.HouseStyle.replace({'1.5Unf':1,
    12                                           '1.5Fin':2, '2.5Unf':2,↳
↳'SFoyer':2,
    13                                           '1Story':3, 'SLvl':3,

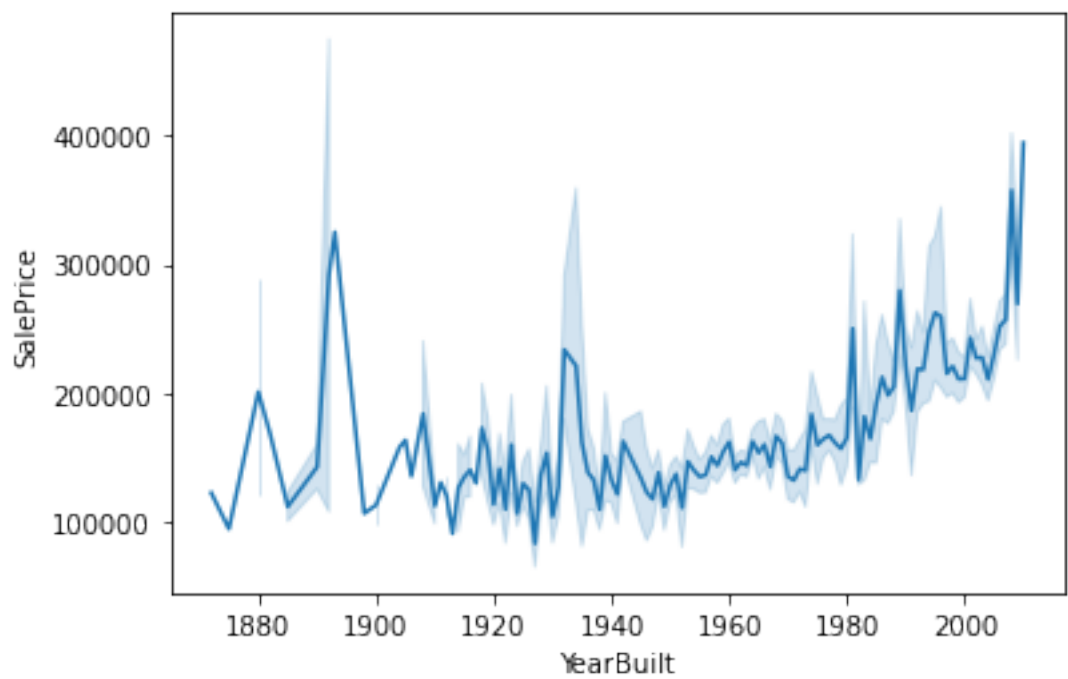
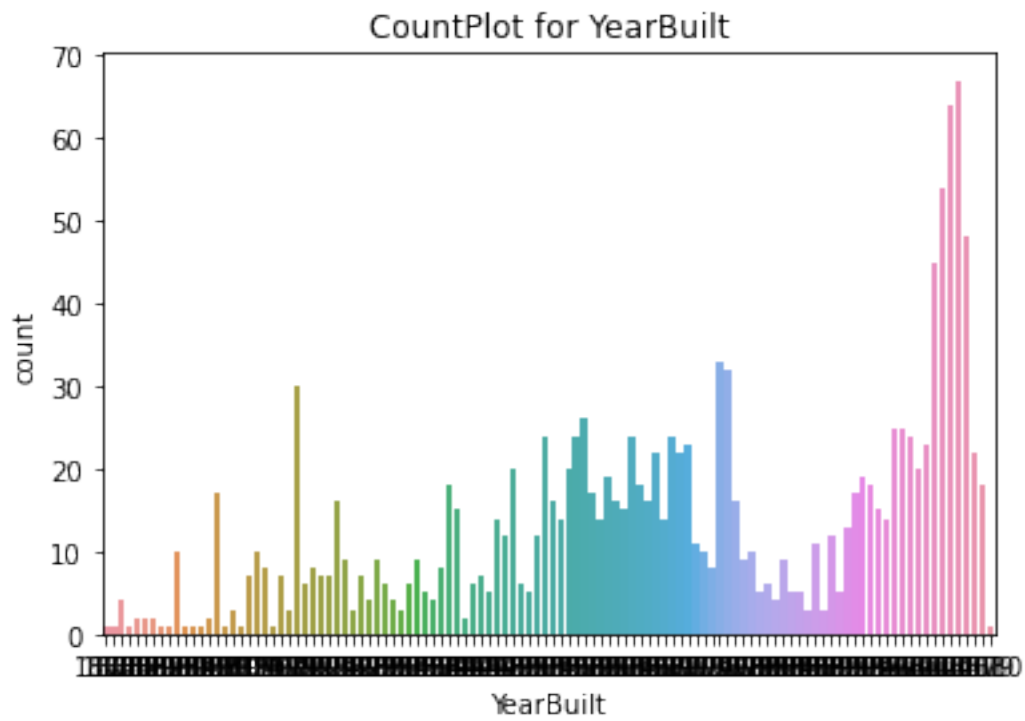
NameError: name 'test' is not defined

```



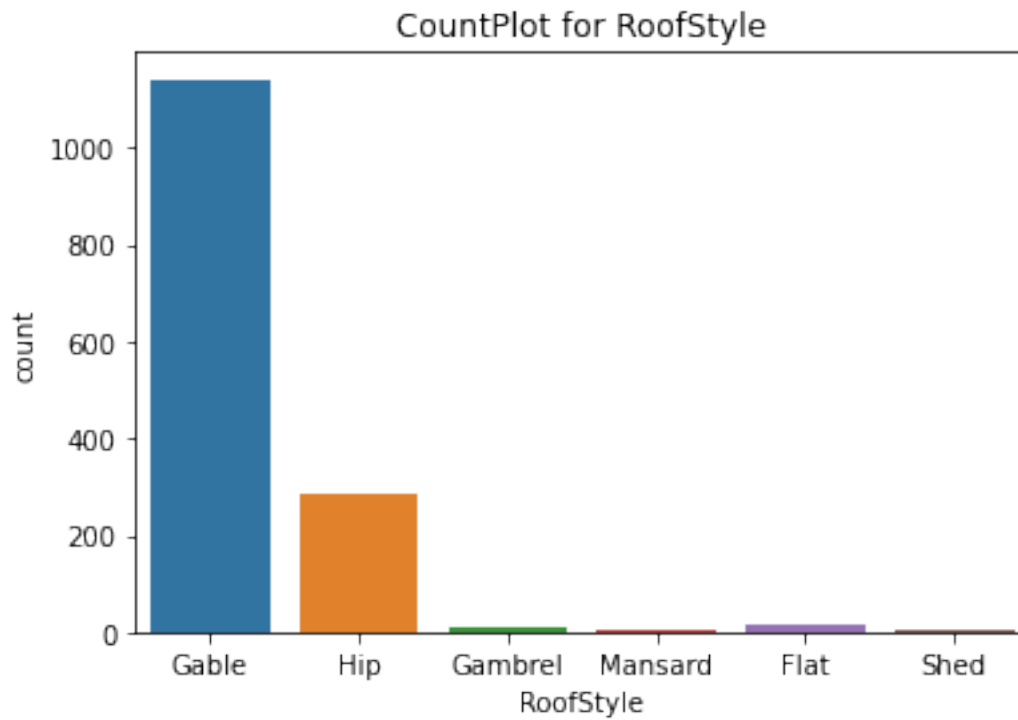
```
[27]: plotting('YearBuilt')
plt.show()
sns.lineplot(y='SalePrice',x='YearBuilt',data=train)
# It is clearly visible. So it should be part of our features variable.
features.append('YearBuilt')
```

No. of missing values are :
0

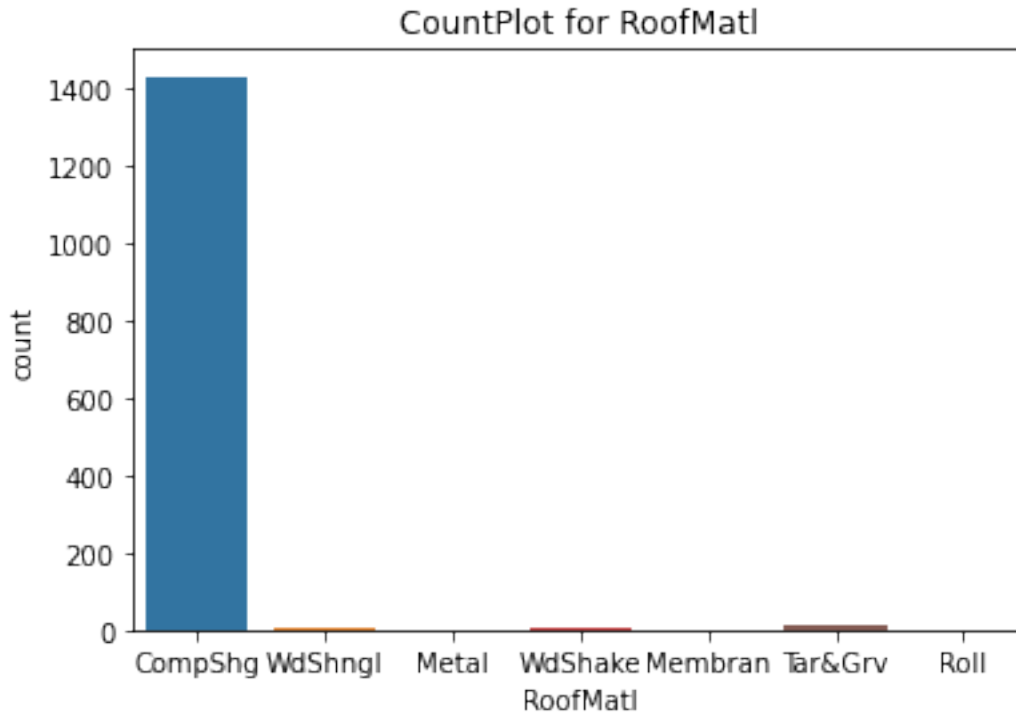


```
[28]: #Roof Style : Type of roof
      plotting('RoofStyle')
      plt.show()
      #RoofMatl: Roof material
      plotting('RoofMatl')
```

No. of missing values are :
0



No. of missing values are :
0



```
[29]: plotting('Exterior1st')
plt.show()
#Exterior2nd: Exterior covering on house (if more than one material)
plotting('Exterior2nd')
plt.show()

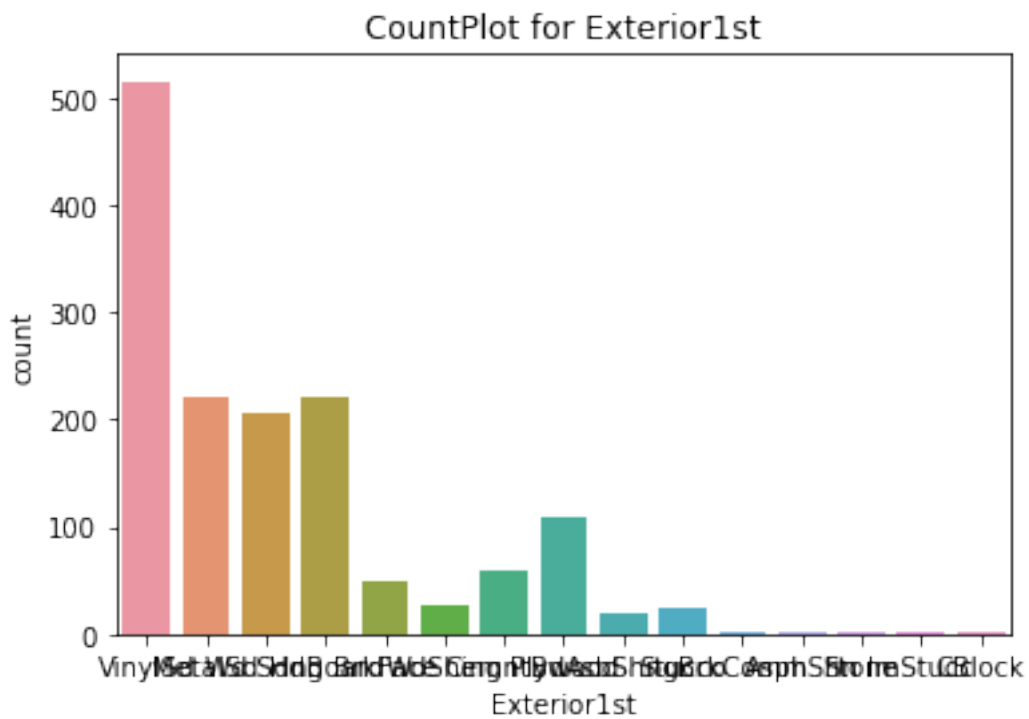
train.Exterior1st=train.Exterior1st.replace({'BrkComm':1,
                                              'AsphShn':2, 'CBlock':2, 'AsbShng':
↪2,
                                              'WdShing':3, 'Wd Sdng':3,
↪'MetalSd':3, 'Stucco':3, 'HdBoard':3,
                                              'BrkFace':4, 'Plywood':4,
                                              'VinylSd':5,
                                              'CemntBd':6,
                                              'Stone':7, 'ImStucc':7})

test.Exterior1st=test.Exterior1st.replace({'BrkComm':1,
                                             'AsphShn':2, 'CBlock':2, 'AsbShng':
↪2,
                                             'WdShing':3, 'Wd Sdng':3,
↪'MetalSd':3, 'Stucco':3, 'HdBoard':3,
                                             'BrkFace':4, 'Plywood':4,
                                             'VinylSd':5,
```

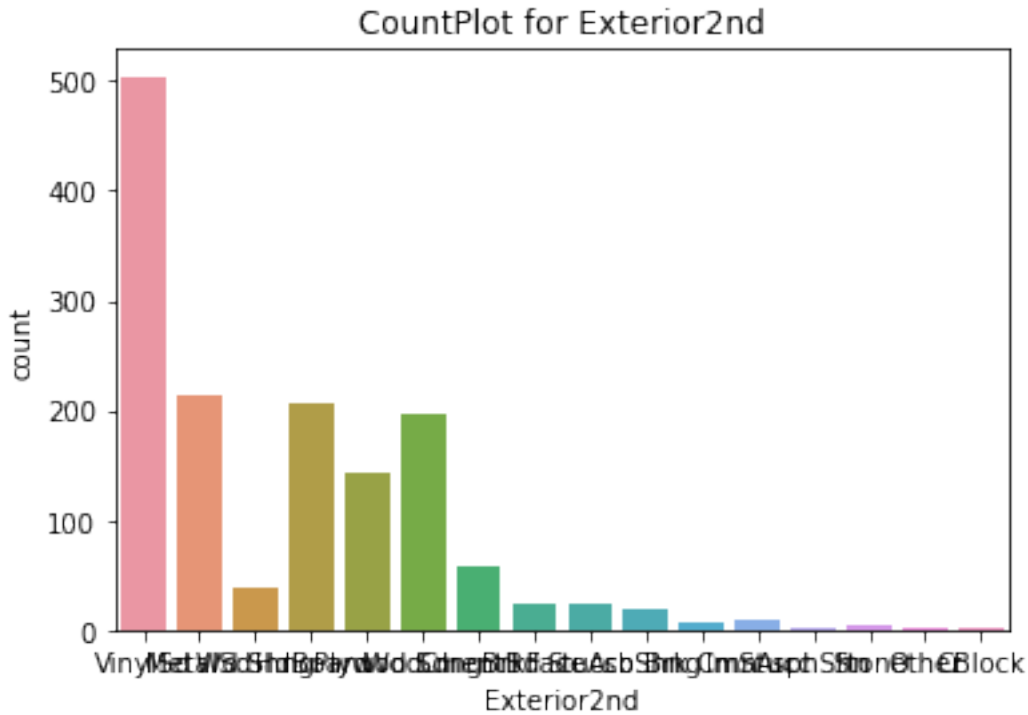
```
'CemntBd':6,  
'Stone':7, 'ImStucc':7})
```

```
features.append('Exterior1st')
```

No. of missing values are :
0



No. of missing values are :
0



```

NameError                                Traceback (most recent call
↳ last)

<ipython-input-29-10b24e7acb39> in <module>
    13                                     'Stone':7, 'ImStucc':7})
    14
--> 15 test.Exterior1st=test.Exterior1st.replace({'BrkComm':1,
    16                                             'AsphShn':2, 'CBlock':
↳ 2, 'AsbShng':2,
    17                                             'WdShng':3, 'Wd Sdng':
↳ 3, 'MetalSd':3, 'Stucco':3, 'HdBoard':3,

NameError: name 'test' is not defined

```

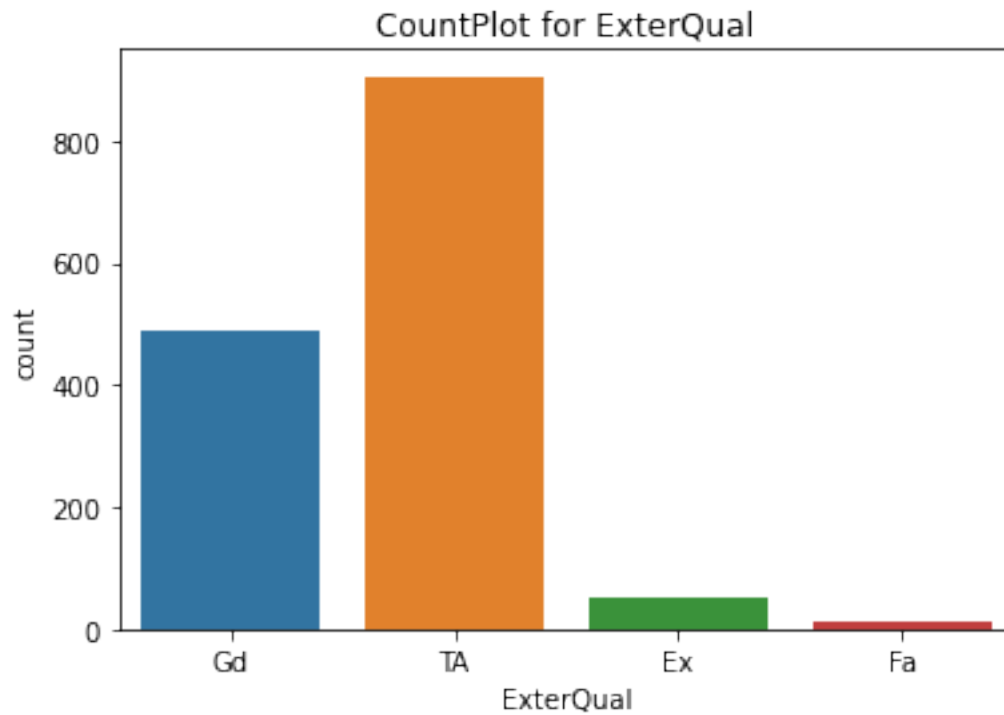
```

[30]: plotting('ExterQual')
      plt.show()
      sns.lineplot(y='SalePrice',x='ExterQual',data=train)

```

```
features.append('ExterQual')
train.ExterQual = train.ExterQual.replace({'Fa':1, 'TA':2, 'Gd':3, 'Ex':4})
test.ExterQual = test.ExterQual.replace({'Fa':1, 'TA':2, 'Gd':3, 'Ex':4})
```

No. of missing values are :
0



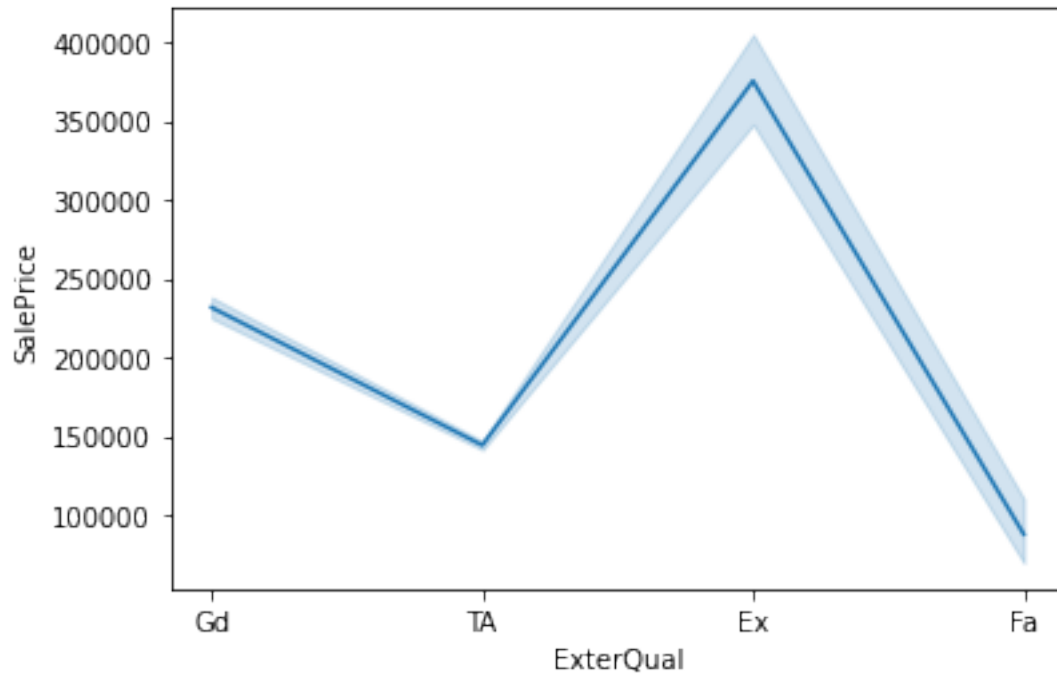
```

↳
-----
NameError                                Traceback (most recent call↳
↳last)

<ipython-input-30-307bfc335c12> in <module>
      4 features.append('ExterQual')
      5 train.ExterQual = train.ExterQual.replace({'Fa':1, 'TA':2, 'Gd':3,↳
↳'Ex':4})
----> 6 test.ExterQual = test.ExterQual.replace({'Fa':1, 'TA':2, 'Gd':3,↳
↳'Ex':4})

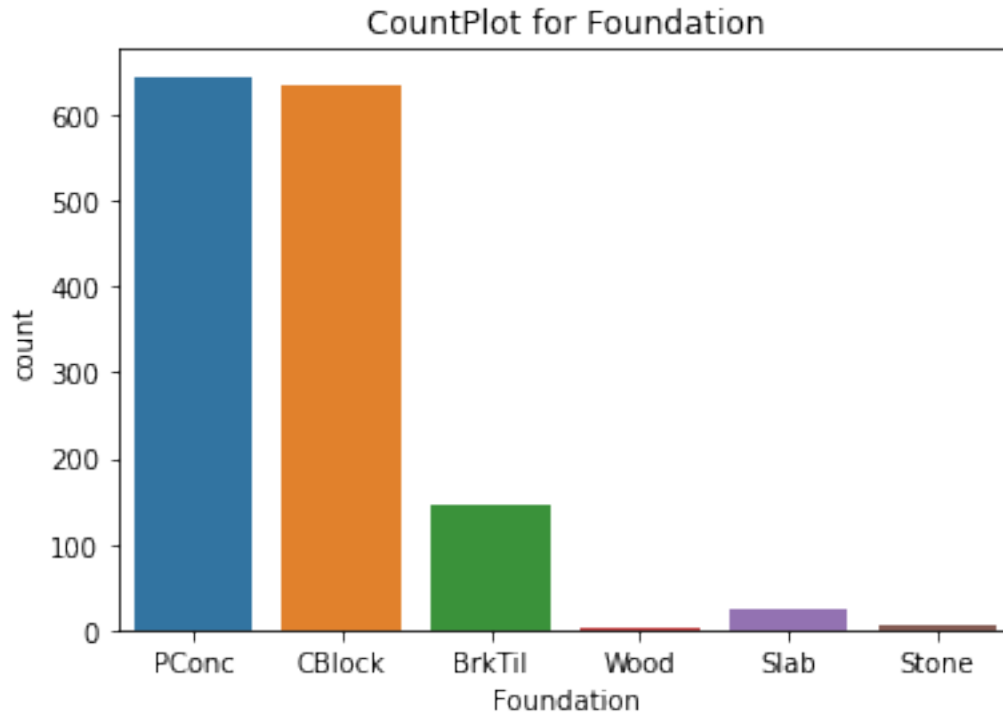
```

NameError: name 'test' is not defined

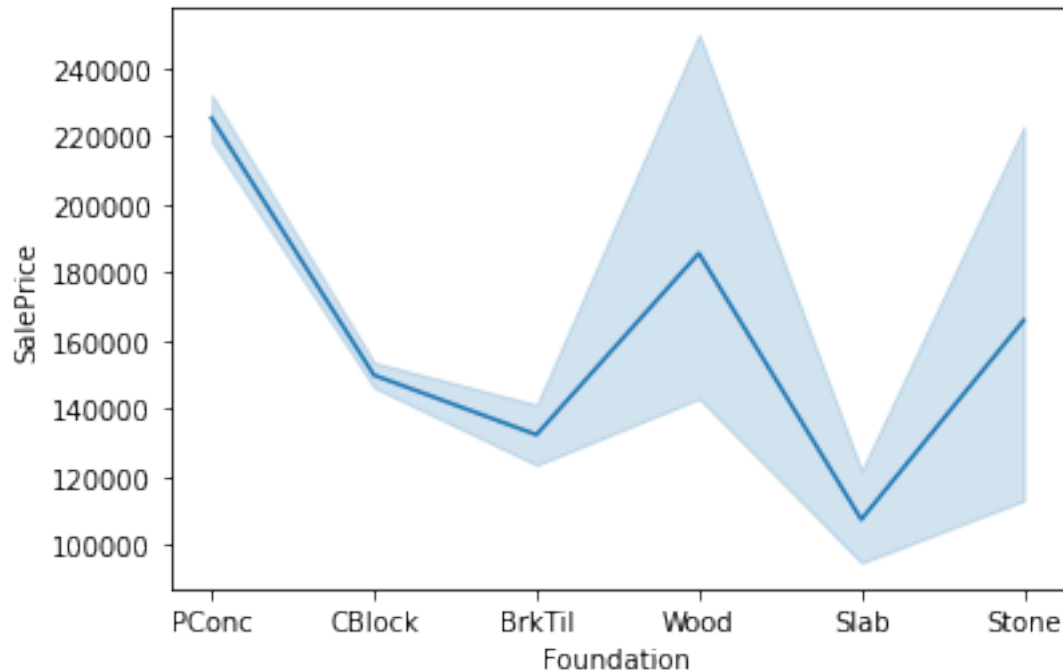


```
[31]: plotting('Foundation')
plt.show()
sns.lineplot(y='SalePrice',x='Foundation',data=train)
train.Foundation=train.Foundation.replace({'Slab':1,
                                           'BrkTil':2, 'CBlock':2, 'Stone':2,
                                           'Wood':3, 'PConc':4})
test.Foundation=test.Foundation.replace({'Slab':1,
                                         'BrkTil':2, 'CBlock':2, 'Stone':2,
                                         'Wood':3, 'PConc':4})
features.append('Foundation')
```

No. of missing values are :
0



```
↳  
-----  
↳  
NameError                                Traceback (most recent call↳  
↳last)  
  
    <ipython-input-31-a223a5f13fd1> in <module>  
        5                                'BrkTil':2, 'CBlock':2,↳  
↳'Stone':2,                                'Wood':3, 'PConc':4})  
        6                                'BrkTil':2, 'CBlock':2,↳  
----> 7 test.Foundation=test.Foundation.replace({'Slab':1,  
        8                                'Wood':3, 'PConc':4})  
↳'Stone':2,                                'Wood':3, 'PConc':4})  
        9  
  
NameError: name 'test' is not defined
```



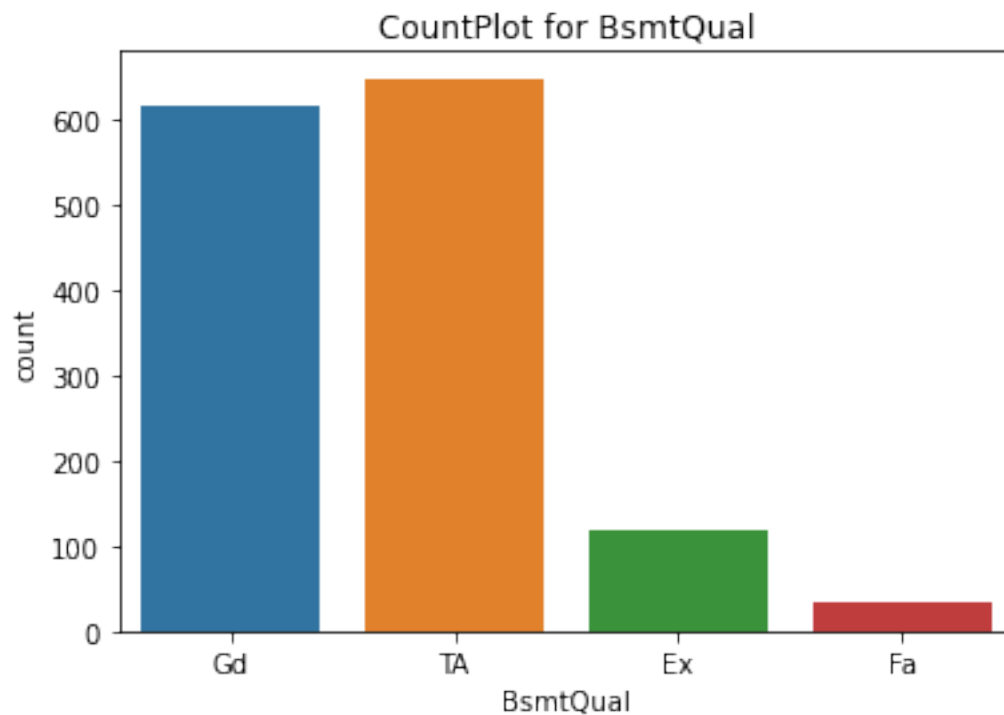
```
[32]: plotting('BsmtQual')
train.BsmtQual=train.BsmtQual.fillna('None').replace({'Fa':2, 'None':1, 'TA':3,
↳ 'Gd':4, 'Ex':5})
test.BsmtQual=test.BsmtQual.fillna('None').replace({'Fa':2, 'None':1, 'TA':3,
↳ 'Gd':4, 'Ex':5})
features.append('BsmtQual')
```

No. of missing values are :
37

```
↳
-----
NameError                                Traceback (most recent call
↳ last)

<ipython-input-32-fa674785bf39> in <module>
      1 plotting('BsmtQual')
      2 train.BsmtQual=train.BsmtQual.fillna('None').replace({'Fa':2, 'None':
↳ 1, 'TA':3, 'Gd':4, 'Ex':5})
----> 3 test.BsmtQual=test.BsmtQual.fillna('None').replace({'Fa':2, 'None':
↳ 1, 'TA':3, 'Gd':4, 'Ex':5})
      4 features.append('BsmtQual')
```

NameError: name 'test' is not defined



```
[33]: plotting('BsmtExposure')
train.BsmtExposure=train.BsmtExposure.fillna('None').replace({'None':1, 'No':2,
↳ 'Av':3, 'Mn':3, 'Gd':4})
test.BsmtExposure=test.BsmtExposure.fillna('None').replace({'None':1, 'No':2,
↳ 'Av':3, 'Mn':3, 'Gd':4})
features.append('BsmtExposure')
```

No. of missing values are :
38

```
↳
↳ -----
NameError                                Traceback (most recent call↳
↳ last)

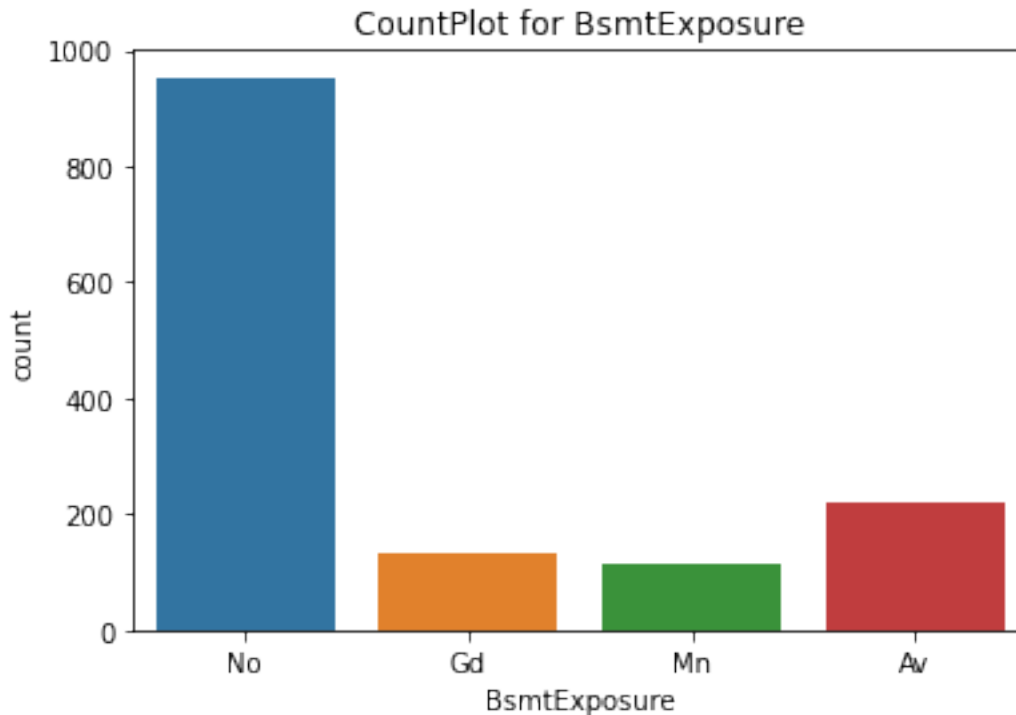
<ipython-input-33-1a14666ffba7> in <module>
    1 plotting('BsmtExposure')
```

```

2 train.BsmtExposure=train.BsmtExposure.fillna('None').replace({'None':
↳1, 'No':2, 'Av':3, 'Mn':3, 'Gd':4})
----> 3 test.BsmtExposure=test.BsmtExposure.fillna('None').replace({'None':
↳1, 'No':2, 'Av':3, 'Mn':3, 'Gd':4})
4 features.append('BsmtExposure')

```

NameError: name 'test' is not defined



```

[34]: plotting('HeatingQC')
train.HeatingQC.replace({'Po':1, 'Fa':2, 'TA':3, 'Gd':4, 'Ex':5},inplace=True)
test.HeatingQC.replace({'Po':1, 'Fa':2, 'TA':3, 'Gd':4, 'Ex':5},inplace=True)
features.append('HeatingQC')

```

No. of missing values are :
0

```

↳ -----
NameError                                Traceback (most recent call↳
↳last)

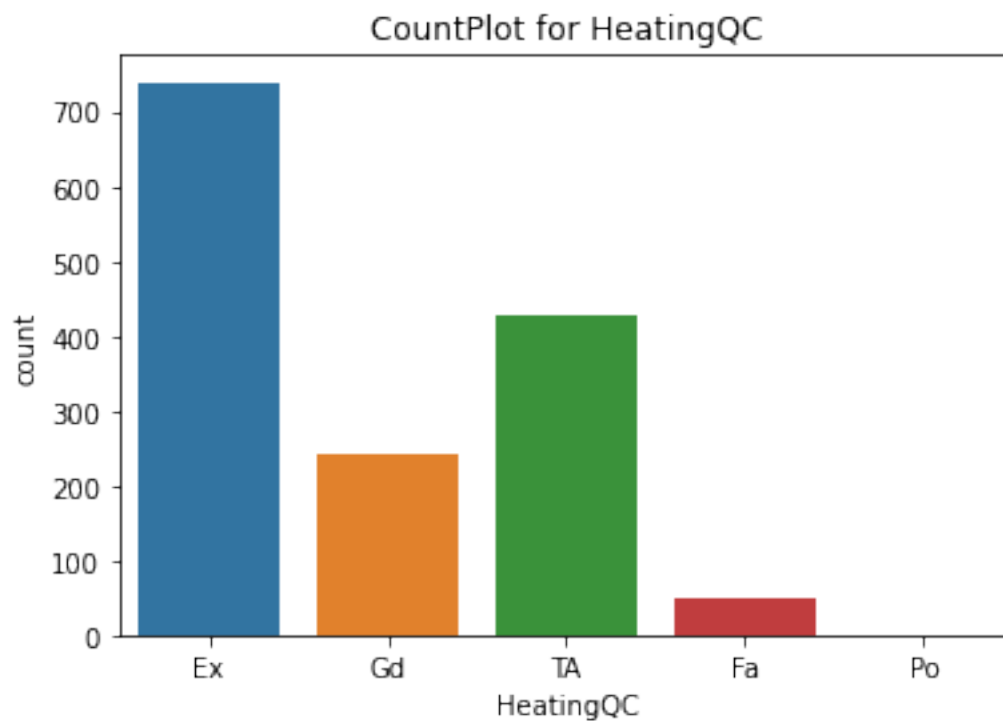
```

```

<ipython-input-34-5239a93c58d1> in <module>
    1 plotting('HeatingQC')
    2 train.HeatingQC.replace({'Po':1, 'Fa':2, 'TA':3, 'Gd':4, 'Ex':
↪5},inplace=True)
----> 3 test.HeatingQC.replace({'Po':1, 'Fa':2, 'TA':3, 'Gd':4, 'Ex':
↪5},inplace=True)
    4 features.append('HeatingQC')

```

NameError: name 'test' is not defined



```

[35]: plotting('KitchenQual')
train.KitchenQual.replace({'Fa':1, 'TA':2, 'Gd':3, 'Ex':4},inplace=True)
test.KitchenQual.replace({'Fa':1, 'TA':2, 'Gd':3, 'Ex':4},inplace=True)
features.append('KitchenQual')

```

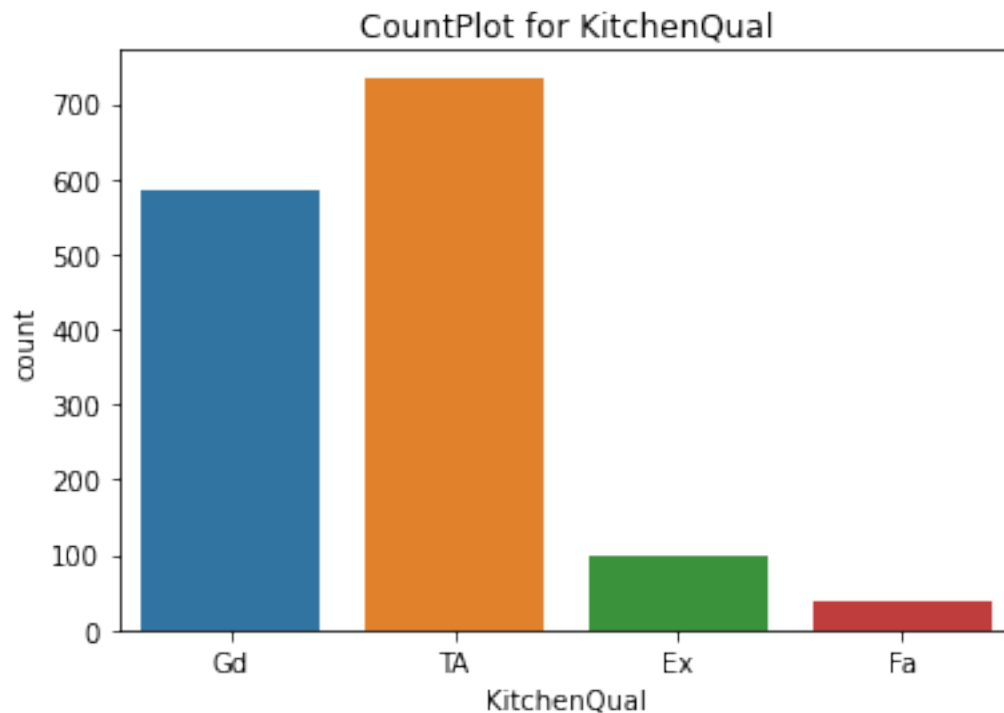
No. of missing values are :
0

↪ -----


```
NameError                                Traceback (most recent call_
↳last)
```

```
<ipython-input-35-86ba227d35ee> in <module>
    1 plotting('KitchenQual')
    2 train.KitchenQual.replace({'Fa':1, 'TA':2, 'Gd':3, 'Ex':
↳4},inplace=True)
----> 3 test.KitchenQual.replace({'Fa':1, 'TA':2, 'Gd':3, 'Ex':
↳4},inplace=True)
    4 features.append('KitchenQual')
```

NameError: name 'test' is not defined



```
[36]: plotting('FireplaceQu')
train.FireplaceQu.fillna('None',inplace=True)
train.FireplaceQu.replace({'None':1, 'Po':1, 'Fa':2, 'TA':3, 'Gd':4, 'Ex':
↳5},inplace=True)
test.FireplaceQu.fillna('None',inplace=True)
test.FireplaceQu.replace({'None':1, 'Po':1, 'Fa':2, 'TA':3, 'Gd':4, 'Ex':
↳5},inplace=True)
```

```
features.append('FireplaceQu')
```

No. of missing values are :
690

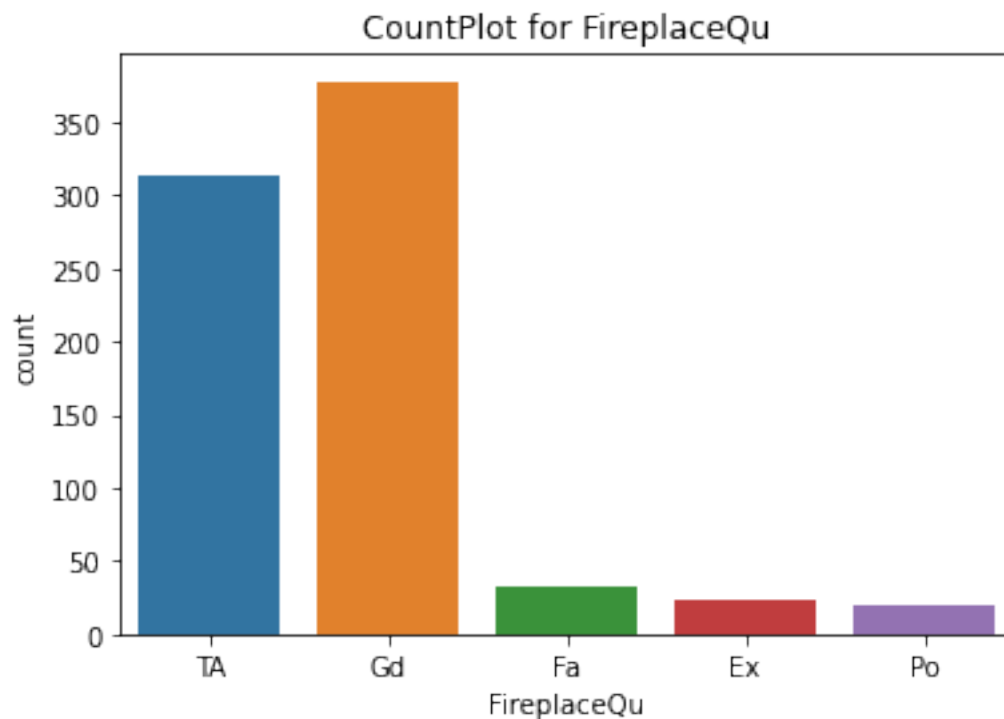
```

      □
↳ -----

NameError                                Traceback (most recent call↳
↳ last)

<ipython-input-36-3bd8e3a474b3> in <module>
      2 train.FireplaceQu.fillna('None',inplace=True)
      3 train.FireplaceQu.replace({'None':1, 'Po':1, 'Fa':2, 'TA':3, 'Gd':4,↳
↳ 'Ex':5},inplace=True)
----> 4 test.FireplaceQu.fillna('None',inplace=True)
      5 test.FireplaceQu.replace({'None':1, 'Po':1, 'Fa':2, 'TA':3, 'Gd':4,↳
↳ 'Ex':5},inplace=True)
      6 features.append('FireplaceQu')
```

NameError: name 'test' is not defined



```
[37]: plotting('GarageType')
train.GarageType.fillna('None',inplace=True)
test.GarageType.fillna('None',inplace=True)
train.GarageType.replace({'CarPort':1, 'None':1,
                           'Detchd':2,
                           '2Types':3, 'Basment':3,
                           'Attchd':4, 'BuiltIn':
→5},inplace=True)
test.GarageType.replace({'CarPort':1, 'None':1,
                           'Detchd':2,
                           '2Types':3, 'Basment':3,
                           'Attchd':4, 'BuiltIn':
→5},inplace=True)
features.append('GarageType')
```

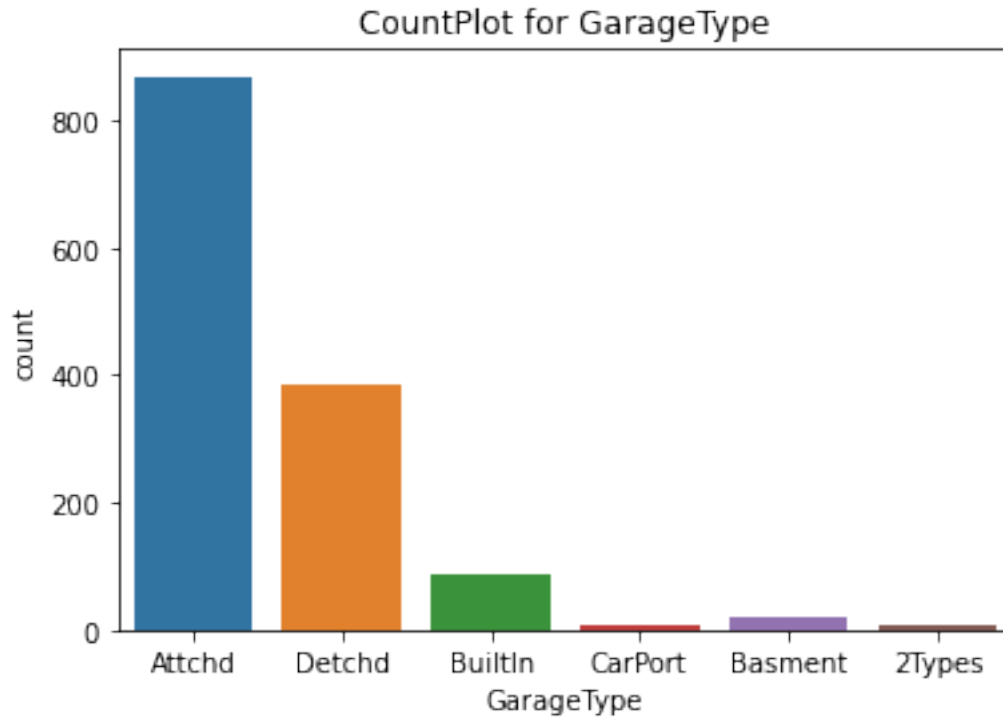
No. of missing values are :
81

```
↳ -----
```

```
NameError                                Traceback (most recent call↳
↳last)
```

```
<ipython-input-37-8da765d6e288> in <module>
    1 plotting('GarageType')
    2 train.GarageType.fillna('None',inplace=True)
----> 3 test.GarageType.fillna('None',inplace=True)
    4 train.GarageType.replace({'CarPort':1, 'None':1,
    5                             'Detchd':2,
```

NameError: name 'test' is not defined



```
[38]: plotting('GarageFinish')
train.GarageFinish.fillna('None',inplace=True)
test.GarageFinish.fillna('None',inplace=True)
train.GarageFinish.replace({'None':1, 'Unf':2, 'RFn':3, 'Fin':4},inplace=True)
test.GarageFinish.replace({'None':1, 'Unf':2, 'RFn':3, 'Fin':4},inplace=True)
features.append('GarageFinish')
```

No. of missing values are :
81

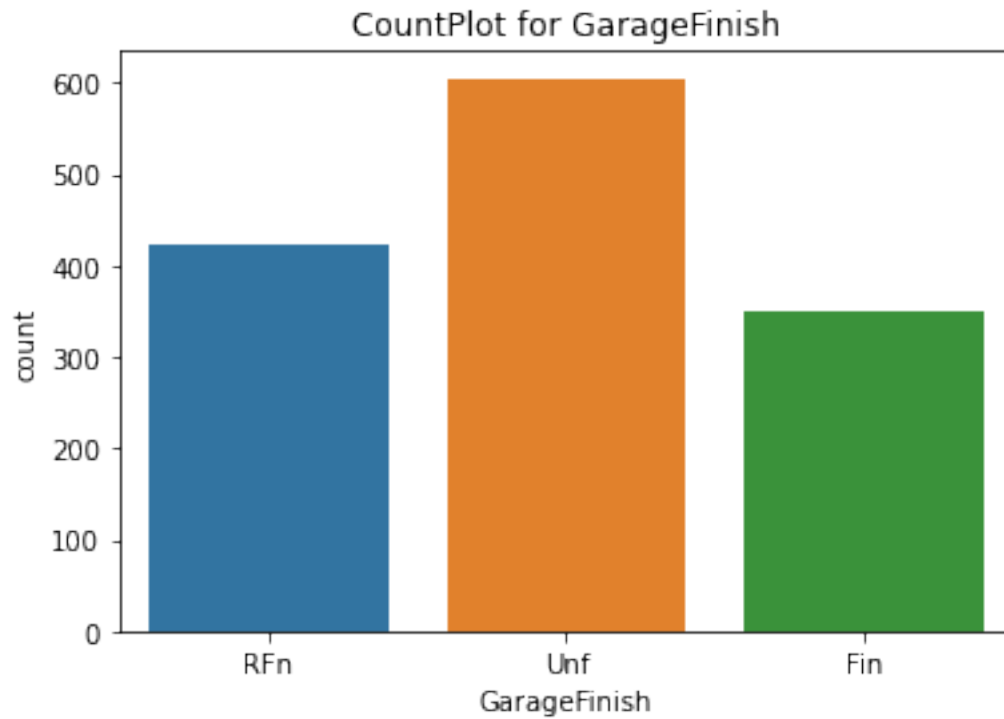
```

↳
-----
NameError                                Traceback (most recent call↳
↳last)

<ipython-input-38-0f0163201bfb> in <module>
    1 plotting('GarageFinish')
    2 train.GarageFinish.fillna('None',inplace=True)
----> 3 test.GarageFinish.fillna('None',inplace=True)
       4 train.GarageFinish.replace({'None':1, 'Unf':2, 'RFn':3, 'Fin':
↳4},inplace=True)
```

```
5 test.GarageFinish.replace({'None':1, 'Unf':2, 'RFn':3, 'Fin':
↳4},inplace=True)
```

NameError: name 'test' is not defined



```
[39]: full=pd.concat([train,test],ignore_index=True)
full.drop('Id',axis=1,inplace=True)
full.shape
```

```

↳
-----
NameError                                Traceback (most recent call↳
↳last)

<ipython-input-39-c1256a1135a3> in <module>
----> 1 full=pd.concat([train,test],ignore_index=True)
      2 full.drop('Id',axis=1,inplace=True)
      3 full.shape
```

```
NameError: name 'test' is not defined
```

```
[40]: full.isnull().sum().sort_values(ascending=False)
```

```
↳
-----
NameError                                Traceback (most recent call↳
↳last)

<ipython-input-40-02d623efd052> in <module>
----> 1 full.isnull().sum().sort_values(ascending=False)
```

```
NameError: name 'full' is not defined
```

```
[41]: train.groupby(['Neighborhood'])[['LotFrontage']].agg(['mean', 'median', 'count'])
```

```
[41]:
```

	LotFrontage		
	mean	median	count
Neighborhood			
1	27.800000	21.0	15
2	49.400000	50.0	50
3	62.440000	60.0	250
4	68.164384	66.0	73
5	63.930233	69.0	43
6	77.409722	76.0	144
7	70.726804	70.0	194
8	68.400000	75.0	115
9	62.700000	61.5	20
10	84.908257	90.0	109
mes	76.462366	73.0	186

```
[43]: train['LotAreaCut'] = pd.qcut(train.LotArea,10)

train.groupby([train['LotAreaCut']])[['LotFrontage']].
↳agg(['mean', 'median', 'count'])
```

```
[43]:
```

	LotFrontage		
	mean	median	count
LotAreaCut			
(1299.999, 5000.0]	36.691057	35.0	123
(5000.0, 7071.2]	55.984252	52.0	127
(7071.2, 8063.1]	63.869919	61.0	123

(8063.1, 8788.8]	67.143939	65.0	132
(8788.8, 9475.0]	70.544715	70.0	123
(9475.0, 10193.0]	75.195312	76.0	128
(10193.0, 11049.9]	73.780488	75.0	123
(11049.9, 12194.6]	84.649123	84.5	114
(12194.6, 14340.9]	84.733945	85.0	109
(14340.9, 215245.0]	93.628866	90.0	97

```
[44]: train['LotFrontage']= train.
      ↳groupby(['LotAreaCut','Neighborhood'])['LotFrontage'].transform(lambda x : x.
      ↳fillna(x.median()))
```

```
[47]: columns = ["MasVnrArea", "BsmtUnfSF", "TotalBsmtSF", "GarageCars",
      ↳"BsmtFinSF2", "BsmtFinSF1", "GarageArea"]
      for col in columns:
          train[col].fillna(0,inplace= True)
```

```
[48]: columns1 = ["PoolQC" , "MiscFeature", "Alley", "Fence", "FireplaceQu",
      ↳"GarageQual", "GarageCond", "GarageFinish",
      "GarageYrBlt", "GarageType", "BsmtExposure", "BsmtCond", "BsmtQual",
      ↳"BsmtFinType2", "BsmtFinType1", "MasVnrType"]
      for col1 in columns1:
          train[col1].fillna('None',inplace = True)
```

```
[49]: columns2 = ["MSZoning", "BsmtFullBath", "BsmtHalfBath", "Utilities",
      ↳"Functional",
      "Electrical", "KitchenQual", "SaleType","Exterior1st",
      ↳"Exterior2nd"]
      for col2 in columns2:
          train[col2].fillna(full[col2].mode()[0],inplace = True)
```

```

      ↳
      ↳-----
      ↳
      ↳NameError                                Traceback (most recent call
      ↳last)

      <ipython-input-49-768063900bd8> in <module>
          3
          4 for col2 in columns2:
----> 5     train[col2].fillna(full[col2].mode()[0],inplace = True)

      NameError: name 'full' is not defined
```

```
[50]: train.drop("LotAreaCut",axis=1,inplace=True)
```

```
[51]: features
```

```
[51]: ['LotShape',  
      'LotConfig',  
      'Neighborhood',  
      'HouseStyle',  
      'YearBuilt',  
      'ExterQual']
```

```
[52]: cols_num =  
      → ["MSSubClass", "BsmtFullBath", "BsmtHalfBath", "HalfBath", "BedroomAbvGr", "KitchenAbvGr", "MoSold",  
         "YrSold", "YearBuilt", "YearRemodAdd", "LowQualFinSF", "GarageYrBlt"]  
  
for i in cols_num :  
    train[i]=train[i].astype(str)
```

```
↳ -----  
  
      KeyError                                Traceback (most recent call↳  
↳ last)  
  
    /usr/local/lib/python3.7/site-packages/pandas/core/indexes/base.py in↳  
↳ get_loc(self, key, method, tolerance)  
    2897         try:  
-> 2898             return self._engine.get_loc(casted_key)  
    2899         except KeyError as err:  
  
    pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()  
  
    pandas/_libs/index.pyx in pandas._libs.index.IndexEngine.get_loc()  
  
    pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.  
↳ PyObjectHashTable.get_item()  
  
    pandas/_libs/hashtable_class_helper.pxi in pandas._libs.hashtable.  
↳ PyObjectHashTable.get_item()  
  
KeyError: 'KitchenAbvGr'
```


The above exception was the direct cause of the following exception:

```
KeyError                                Traceback (most recent call
↳ last)

<ipython-input-52-48c69d59c9e4> in <module>
      3
      4 for i in cols_num :
----> 5     train[i]=train[i].astype(str)

/usr/local/lib/python3.7/site-packages/pandas/core/frame.py in
↳ __getitem__(self, key)
    2904         if self.columns.nlevels > 1:
    2905             return self._getitem_multilevel(key)
-> 2906         indexer = self.columns.get_loc(key)
    2907         if is_integer(indexer):
    2908             indexer = [indexer]

/usr/local/lib/python3.7/site-packages/pandas/core/indexes/base.py in
↳ get_loc(self, key, method, tolerance)
    2898         return self._engine.get_loc(casted_key)
    2899         except KeyError as err:
-> 2900             raise KeyError(key) from err
    2901
    2902         if tolerance is not None:
```

KeyError: 'KitchenAbvGr'

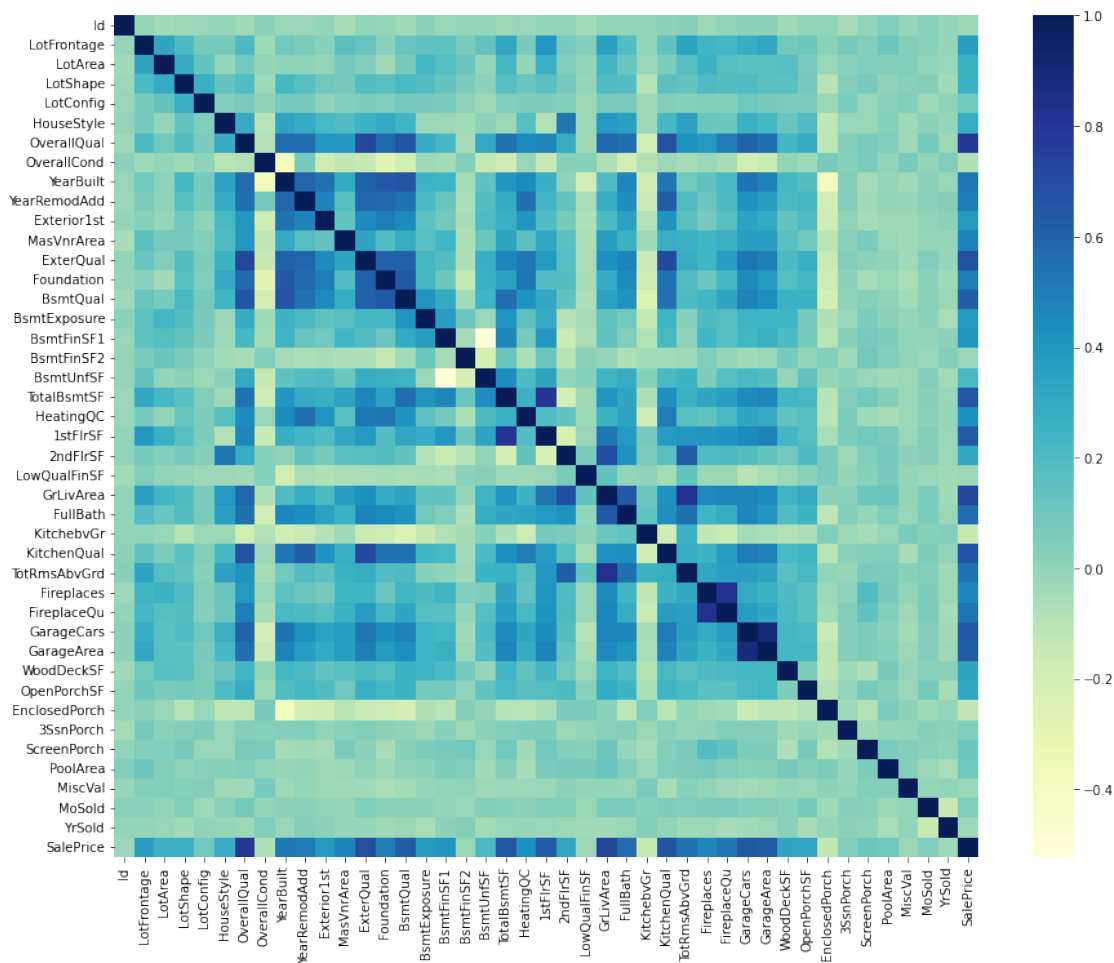
```
[53]: features
```

```
[53]: ['LotShape',
      'LotConfig',
      'Neighborhood',
      'HouseStyle',
      'YearBuilt',
      'ExterQual']
```

```
[54]: train.isnull().sum(0).sort_values(ascending=False)
```

```
[54]: Electrical      1
      SalePrice       0
      Foundation     0
      RoofMatl       0
      Exterior1st    0
      ..
      Functiol       0
      Fireplaces     0
      FireplaceQu    0
      GarageType     0
      Id            0
      Length: 81, dtype: int64
```

```
[55]: #Checking correlation via heatmap:
plt.figure(figsize=(15,12))
sns.heatmap(train.corr(), cmap="YlGnBu")
plt.show()
```



```
[56]: core=train.corr()
      core_des=core.sort_values(by='SalePrice',ascending=False)
      core_des
```

```
[56]:
```

	Id	LotFrontage	LotArea	LotShape	LotConfig	\
SalePrice	-0.021673	0.361292	0.268179	0.270477	0.092499	
OverallQual	-0.030213	0.206488	0.091392	0.190817	0.064424	
GrLivArea	0.003138	0.353064	0.232927	0.192468	0.060986	
ExterQual	0.003413	0.122734	0.040373	0.180443	0.051259	
KitchenQual	0.013709	0.140458	0.055988	0.151122	0.034402	
TotalBsmtSF	-0.025908	0.313779	0.223914	0.172605	0.042197	
GarageCars	0.016863	0.281073	0.152484	0.194461	0.097318	
1stFlrSF	0.003921	0.394293	0.269234	0.165273	0.048999	
GarageArea	0.014316	0.317903	0.163680	0.161068	0.068097	
BsmtQual	-0.033854	0.121623	0.067061	0.217596	0.083792	
FullBath	0.005750	0.187091	0.119679	0.181745	0.060946	
TotRmsAbvGrd	0.025227	0.339434	0.175425	0.126653	0.026261	
FireplaceQu	0.003231	0.229629	0.177234	0.182126	0.030781	
YearBuilt	-0.013463	0.091459	0.007462	0.226674	0.083040	
YearRemodAdd	-0.022694	0.056648	0.007500	0.172670	0.075750	
Foundation	-0.007603	0.023025	-0.037182	0.165594	0.047767	
MasVnrArea	-0.053952	0.152038	0.083715	0.076833	0.042114	
Fireplaces	-0.023122	0.239627	0.261176	0.192602	0.041528	
HeatingQC	-0.015546	0.070610	-0.001521	0.117117	0.050443	
BsmtFinSF1	-0.015012	0.146506	0.175320	0.127540	0.064786	
Exterior1st	-0.001654	0.027434	-0.008814	0.100561	0.009628	
BsmtExposure	0.015689	0.149516	0.229513	0.182028	0.069130	
LotFrontage	-0.010181	1.000000	0.323789	0.204337	0.074351	
WoodDeckSF	-0.030262	0.092277	0.168621	0.159403	0.056044	
OpenPorchSF	-0.002115	0.105699	0.061861	0.079791	0.059458	
2ndFlrSF	0.005248	0.070666	0.041026	0.084127	0.031950	
HouseStyle	-0.007124	0.077233	-0.006970	0.127657	0.044283	
LotShape	-0.028067	0.204337	0.303146	1.000000	0.276815	
LotArea	-0.037689	0.323789	1.000000	0.303146	0.131158	
BsmtUnfSF	-0.007499	0.127331	-0.003251	0.014493	-0.027797	
ScreenPorch	0.001491	0.042183	0.045288	0.066678	-0.016082	
PoolArea	0.048476	0.106658	0.036491	0.011531	0.024744	
LotConfig	-0.015053	0.074351	0.131158	0.276815	1.000000	
MoSold	0.023511	0.022187	0.005744	0.031968	-0.023353	
3SsnPorch	-0.046600	0.067284	0.021362	0.034485	0.058946	
BsmtFinSF2	-0.005802	0.064977	0.114297	0.061950	0.010835	
MiscVal	-0.006195	0.035609	0.039072	0.029176	0.031796	
Id	1.000000	-0.010181	-0.037689	-0.028067	-0.015053	
LowQualFinSF	-0.044191	0.038046	0.005536	-0.007576	-0.024001	
YrSold	0.000378	0.000349	-0.013699	-0.037661	0.011064	
OverallCond	0.012925	-0.029367	-0.002723	-0.032042	0.012180	
EnclosedPorch	0.003103	0.017687	-0.016505	-0.094430	-0.021226	

KitchenbvGr	0.003078	0.001710	-0.016796	-0.094759	-0.025386	
	HouseStyle	OverallQual	OverallCond	YearBuilt	YearRemodAdd	\
SalePrice	0.292263	0.795774	-0.077948	0.523608	0.507717	
OverallQual	0.295311	1.000000	-0.090442	0.571368	0.549827	
GrLivArea	0.387912	0.589414	-0.077423	0.194397	0.286758	
ExterQual	0.252170	0.723286	-0.137724	0.597356	0.586675	
KitchenQual	0.224432	0.670627	-0.025207	0.528822	0.624563	
TotalBsmtSF	-0.028472	0.538595	-0.174931	0.400266	0.294941	
GarageCars	0.272117	0.600741	-0.185275	0.537301	0.419882	
1stFlrSF	-0.086386	0.469092	-0.144717	0.281253	0.238769	
GarageArea	0.221161	0.557230	-0.150503	0.477998	0.370000	
BsmtQual	0.241591	0.658657	-0.200155	0.656603	0.546599	
FullBath	0.354523	0.548420	-0.193473	0.467146	0.437927	
TotRmsAbvGrd	0.346882	0.420621	-0.055782	0.091220	0.188418	
FireplaceQu	0.113930	0.497400	-0.055271	0.219047	0.199877	
YearBuilt	0.335103	0.571368	-0.375572	1.000000	0.592105	
YearRemodAdd	0.293811	0.549827	0.074615	0.592105	1.000000	
Foundation	0.307361	0.577360	-0.279920	0.644117	0.573413	
MasVnrArea	0.196764	0.398785	-0.124261	0.308722	0.172590	
Fireplaces	0.116101	0.392472	-0.022406	0.144982	0.109898	
HeatingQC	0.175592	0.456379	-0.013482	0.448075	0.549420	
BsmtFinSF1	-0.024982	0.222467	-0.042987	0.249818	0.123039	
Exterior1st	0.217661	0.396564	-0.172392	0.545668	0.480334	
BsmtExposure	-0.020630	0.295445	-0.031653	0.254469	0.207194	
LotFrontage	0.077233	0.206488	-0.029367	0.091459	0.056648	
WoodDeckSF	0.116652	0.236753	-0.002675	0.223700	0.204591	
OpenPorchSF	0.194471	0.297186	-0.029800	0.184116	0.222790	
2ndFlrSF	0.529462	0.290052	0.030504	0.006968	0.137423	
HouseStyle	1.000000	0.295311	-0.103506	0.335103	0.293811	
LotShape	0.127657	0.190817	-0.032042	0.226674	0.172670	
LotArea	-0.006970	0.091392	-0.002723	0.007462	0.007500	
BsmtUnfSF	0.013420	0.308990	-0.136723	0.148836	0.180958	
ScreenPorch	-0.014635	0.066298	0.054632	-0.049972	-0.038360	
PoolArea	0.027652	0.044758	0.002321	-0.005171	-0.003426	
LotConfig	0.044283	0.064424	0.012180	0.083040	0.075750	
MoSold	0.015159	0.072531	-0.003735	0.012975	0.022020	
3SsnPorch	-0.011804	0.030991	0.025426	0.031578	0.045504	
BsmtFinSF2	-0.042940	-0.058324	0.040034	-0.048683	-0.067377	
MiscVal	-0.033895	-0.031238	0.068728	-0.034273	-0.010160	
Id	-0.007124	-0.030213	0.012925	-0.013463	-0.022694	
LowQualFinSF	-0.028556	-0.030129	0.025414	-0.183770	-0.062289	
YrSold	-0.008630	-0.026575	0.043795	-0.013250	0.036145	
OverallCond	-0.103506	-0.090442	1.000000	-0.375572	0.074615	
EnclosedPorch	-0.121171	-0.113176	0.070120	-0.387093	-0.193551	
KitchenbvGr	-0.024264	-0.184077	-0.087170	-0.174630	-0.149406	

	...	WoodDeckSF	OpenPorchSF	EnclosedPorch	3SsnPorch	\
SalePrice	...	0.324758	0.321142	-0.128646	0.044568	
OverallQual	...	0.236753	0.297186	-0.113176	0.030991	
GrLivArea	...	0.247120	0.302569	0.012976	0.022520	
ExterQual	...	0.205586	0.279707	-0.149025	0.037582	
KitchenQual	...	0.219112	0.249783	-0.105126	0.022769	
TotalBsmtSF	...	0.234476	0.215140	-0.096773	0.040921	
GarageCars	...	0.225462	0.209968	-0.151110	0.035957	
1stFlrSF	...	0.235439	0.178859	-0.064247	0.059521	
GarageArea	...	0.222482	0.228246	-0.121194	0.035880	
BsmtQual	...	0.262482	0.260816	-0.203033	0.010975	
FullBath	...	0.186147	0.252678	-0.114558	0.035715	
TotRmsAbvGrd	...	0.163218	0.220052	0.005700	-0.006235	
FireplaceQu	...	0.211295	0.156796	-0.042915	0.037962	
YearBuilt	...	0.223700	0.184116	-0.387093	0.031578	
YearRemodAdd	...	0.204591	0.222790	-0.193551	0.045504	
Foundation	...	0.176950	0.228337	-0.167918	0.017907	
MasVnrArea	...	0.156812	0.101698	-0.109126	0.019933	
Fireplaces	...	0.198294	0.160924	-0.023879	0.011657	
HeatingQC	...	0.129869	0.172574	-0.101368	0.059826	
BsmtFinSF1	...	0.204993	0.072717	-0.103853	0.029177	
Exterior1st	...	0.126352	0.195974	-0.175786	0.001081	
BsmtExposure	...	0.244759	0.083108	-0.076415	-0.017925	
LotFrontage	...	0.092277	0.105699	0.017687	0.067284	
WoodDeckSF	...	1.000000	0.053686	-0.125616	-0.032642	
OpenPorchSF	...	0.053686	1.000000	-0.092212	-0.005191	
2ndFlrSF	...	0.089902	0.197938	0.063211	-0.024107	
HouseStyle	...	0.116652	0.194471	-0.121171	-0.011804	
LotShape	...	0.159403	0.079791	-0.094430	0.034485	
LotArea	...	0.168621	0.061861	-0.016505	0.021362	
BsmtUnfSF	...	-0.005609	0.128917	-0.002422	0.020806	
ScreenPorch	...	-0.073884	0.077150	-0.083009	-0.031481	
PoolArea	...	0.069100	0.032920	0.060101	-0.007417	
LotConfig	...	0.056044	0.059458	-0.021226	0.058946	
MoSold	...	0.021509	0.072416	-0.029097	0.029483	
3SsnPorch	...	-0.032642	-0.005191	-0.037366	1.000000	
BsmtFinSF2	...	0.068312	0.004974	0.036406	-0.030041	
MiscVal	...	-0.009445	-0.018308	0.018319	0.000340	
Id	...	-0.030262	-0.002115	0.003103	-0.046600	
LowQualFinSF	...	-0.025305	0.019292	0.061028	-0.004315	
YrSold	...	0.022579	-0.056537	-0.010035	0.018610	
OverallCond	...	-0.002675	-0.029800	0.070120	0.025426	
EnclosedPorch	...	-0.125616	-0.092212	1.000000	-0.037366	
KitchenbvGr	...	-0.089916	-0.069810	0.037213	-0.024635	
		ScreenPorch	PoolArea	MiscVal	MoSold	YrSold
SalePrice		0.111415	0.099490	-0.021203	0.046124	-0.028882
						1.000000

OverallQual	0.066298	0.044758	-0.031238	0.072531	-0.026575	0.795774
GrLivArea	0.107768	0.115999	-0.001637	0.057296	-0.035971	0.734968
ExterQual	0.013799	0.012827	-0.039184	0.044280	-0.005866	0.686756
KitchenQual	0.031731	0.046646	-0.054012	0.052985	0.002788	0.662236
TotalBsmtSF	0.092574	0.036089	-0.018514	0.025775	-0.014880	0.651153
GarageCars	0.050909	0.019446	-0.043004	0.039520	-0.038494	0.641047
1stFlrSF	0.095042	0.062756	-0.021045	0.040558	-0.012810	0.631530
GarageArea	0.053015	0.026642	-0.027236	0.032470	-0.027209	0.629217
BsmtQual	0.030462	0.012592	-0.043146	0.016445	-0.017597	0.624758
FullBath	-0.007496	0.045656	-0.014122	0.054756	-0.018697	0.562165
TotRmsAbvGrd	0.060920	0.059728	0.025297	0.039221	-0.033901	0.537769
FireplaceQu	0.147088	0.032945	-0.024111	0.054510	-0.050337	0.527256
YearBuilt	-0.049972	-0.005171	-0.034273	0.012975	-0.013250	0.523608
YearRemodAdd	-0.038360	-0.003426	-0.010160	0.022020	0.036145	0.507717
Foundation	-0.053332	-0.016876	-0.064806	0.021040	-0.033946	0.502768
MasVnrArea	0.064258	-0.020350	-0.029346	-0.005006	-0.007215	0.477810
Fireplaces	0.186238	0.068655	0.001676	0.051142	-0.024315	0.469862
HeatingQC	-0.034613	-0.057745	-0.026780	0.015935	-0.009365	0.428024
BsmtFinSF1	0.068389	0.052981	0.004741	-0.004281	0.015767	0.409384
Exterior1st	-0.049561	-0.025228	-0.024777	0.004053	0.014829	0.383349
BsmtExposure	0.057929	0.013197	-0.005497	0.018078	-0.063932	0.382907
LotFrontage	0.042183	0.106658	0.035609	0.022187	0.000349	0.361292
WoodDeckSF	-0.073884	0.069100	-0.009445	0.021509	0.022579	0.324758
OpenPorchSF	0.077150	0.032920	-0.018308	0.072416	-0.056537	0.321142
2ndFlrSF	0.041504	0.074258	0.016497	0.034665	-0.027775	0.320532
HouseStyle	-0.014635	0.027652	-0.033895	0.015159	-0.008630	0.292263
LotShape	0.066678	0.011531	0.029176	0.031968	-0.037661	0.270477
LotArea	0.045288	0.036491	0.039072	0.005744	-0.013699	0.268179
BsmtUnfSF	-0.012494	-0.034936	-0.023813	0.033997	-0.040953	0.214460
ScreenPorch	1.000000	0.056293	0.031915	0.023151	0.010608	0.111415
PoolArea	0.056293	1.000000	0.031965	-0.018601	-0.064030	0.099490
LotConfig	-0.016082	0.024744	0.031796	-0.023353	0.011064	0.092499
MoSold	0.023151	-0.018601	-0.006544	1.000000	-0.145267	0.046124
3SsnPorch	-0.031481	-0.007417	0.000340	0.029483	0.018610	0.044568
BsmtFinSF2	0.088774	0.046350	0.004905	-0.015362	0.031618	-0.011422
MiscVal	0.031915	0.031965	1.000000	-0.006544	0.004878	-0.021203
Id	0.001491	0.048476	-0.006195	0.023511	0.000378	-0.021673
LowQualFinSF	0.026756	0.066428	-0.003807	-0.022268	-0.028964	-0.025625
YrSold	0.010608	-0.064030	0.004878	-0.145267	1.000000	-0.028882
OverallCond	0.054632	0.002321	0.068728	-0.003735	0.043795	-0.077948
EnclosedPorch	-0.083009	0.060101	0.018319	-0.029097	-0.010035	-0.128646
KitchabvGr	-0.051696	-0.013481	0.062318	0.026554	0.031623	-0.135946

[43 rows x 43 columns]

```
[57]: core_des_top=core_des[(core_des['SalePrice']>0.5) & (core_des['SalePrice'] !=1)]
       core_des_top.index
```

```
[57]: Index(['OverallQual', 'GrLivArea', 'ExterQual', 'KitchenQual', 'TotalBsmtSF',
          'GarageCars', '1stFlrSF', 'GarageArea', 'BsmtQual', 'FullBath',
          'TotRmsAbvGrd', 'FireplaceQu', 'YearBuilt', 'YearRemodAdd',
          'Foundation'],
          dtype='object')
```

```
[58]: features.append(['OverallQual', 'GrLivArea', 'Neighborhood', 'ExterQual',
    ↪ 'KitchenQual',
          'TotalBsmtSF', 'GarageCars', '1stFlrSF', 'GarageArea', 'BsmtQual',
          'FullBath', 'GarageFinish', 'TotRmsAbvGrd', 'FireplaceQu',
          'Foundation'])
```

```
[59]: core_des_low=core_des[(core_des['SalePrice']<0)]
      core_des_low.index
```

```
[59]: Index(['BsmtFinSF2', 'MiscVal', 'Id', 'LowQualFinSF', 'YrSold', 'OverallCond',
          'EnclosedPorch', 'KitchebvGr'],
          dtype='object')
```

```
[60]: features.append(['BsmtFinSF2', 'MiscVal', 'Id', 'OverallCond', 'EnclosedPorch'])
```

```
[61]: features=['LotShape',
          'LotConfig',
          'Neighborhood',
          'HouseStyle',
          'Exterior1st',
          'ExterQual',
          'Foundation',
          'BsmtQual',
          'BsmtExposure',
          'HeatingQC',
          'KitchenQual',
          'FireplaceQu',
          'GarageType',
          'GarageFinish',
          'OverallQual',
          'GrLivArea',
          'Neighborhood',
          'ExterQual',
          'KitchenQual',
          'TotalBsmtSF',
          'GarageCars',
          '1stFlrSF',
          'GarageArea',
          'BsmtQual',
          'FullBath',
          'GarageFinish',
```

```
'TotRmsAbvGrd',
'FireplaceQu',
'Foundation',
'BsmtFinSF2', 'MiscVal', 'Id', 'OverallCond', 'EnclosedPorch', 'SalePrice']
```

```
[62]: train_d=train.copy()
train_d=train_d[features]
```

```
[63]: train_d
```

```
[63]:      LotShape  LotConfig Neighborhood  HouseStyle  Exterior1st  ExterQual  \
0           1           1           7           4           5           3
1           1           4           8           3           3           2
2           2           1           7           4           5           3
3           2           2           7           4           3           2
4           2           4          10           4           5           3
...      ...      ...      ...      ...      ...      ...
1455        1           1           6           4           5           2
1456        1           1           6           3           4           2
1457        1           1           7           4           6           4
1458        1           1          mes           3           3           2
1459        1           1           3           3           3           3
```

```
      Foundation  BsmtQual  BsmtExposure  HeatingQC  ...  GarageFinish  \
0              4          4              2           5  ...          RFn
1              2          4              4           5  ...          RFn
2              4          4              3           5  ...          RFn
3              2          3              2           4  ...          Unf
4              4          4              3           5  ...          RFn
...      ...      ...      ...      ...  ...
1455          4          4              2           5  ...          RFn
1456          2          4              2           3  ...          Unf
1457          2          3              2           5  ...          RFn
1458          2          3              3           4  ...          Unf
1459          2          3              2           4  ...          Fin
```

```
      TotRmsAbvGrd  FireplaceQu  Foundation  BsmtFinSF2  MiscVal  Id  \
0                8            1            4            0            0      1
1                6            3            2            0            0      2
2                6            3            4            0            0      3
3                7            4            2            0            0      4
4                9            3            4            0            0      5
...      ...      ...      ...      ...      ...
1455          7            3            4            0            0  1456
1456          7            3            2          163            0  1457
1457          9            4            2            0          2500  1458
1458          5            1            2          1029            0  1459
```


1459	6	1	2	290	0	1460
------	---	---	---	-----	---	------

	OverallCond	EnclosedPorch	SalePrice
0	5	0	208500
1	8	0	181500
2	5	0	223500
3	5	272	140000
4	5	0	250000
...
1455	5	0	175000
1456	6	0	210000
1457	9	0	266500
1458	6	112	142125
1459	6	0	147500

[1458 rows x 35 columns]

```
[65]: train_d=pd.get_dummies(train_d)
```

```
[66]: train_d.columns
```

```
[66]: Index(['LotShape', 'LotConfig', 'HouseStyle', 'Exterior1st', 'ExterQual',
'Foundation', 'BsmtQual', 'BsmtExposure', 'HeatingQC', 'KitchenQual',
'FireplaceQu', 'OverallQual', 'GrLivArea', 'ExterQual', 'KitchenQual',
'TotalBsmtSF', 'GarageCars', '1stFlrSF', 'GarageArea', 'BsmtQual',
'FullBath', 'TotRmsAbvGrd', 'FireplaceQu', 'Foundation', 'BsmtFinSF2',
'MiscVal', 'Id', 'OverallCond', 'EnclosedPorch', 'SalePrice',
'Neighborhood_1', 'Neighborhood_2', 'Neighborhood_3', 'Neighborhood_4',
'Neighborhood_5', 'Neighborhood_6', 'Neighborhood_7', 'Neighborhood_8',
'Neighborhood_9', 'Neighborhood_10', 'Neighborhood_mes',
'GarageType_2Types', 'GarageType_Attchd', 'GarageType_Basement',
'GarageType_BuiltIn', 'GarageType_CarPort', 'GarageType_Detchd',
'GarageType_None', 'GarageFinish_Fin', 'GarageFinish_None',
'GarageFinish_RFn', 'GarageFinish_Unf', 'Neighborhood_1',
'Neighborhood_2', 'Neighborhood_3', 'Neighborhood_4', 'Neighborhood_5',
'Neighborhood_6', 'Neighborhood_7', 'Neighborhood_8', 'Neighborhood_9',
'Neighborhood_10', 'Neighborhood_mes', 'GarageFinish_Fin',
'GarageFinish_None', 'GarageFinish_RFn', 'GarageFinish_Unf'],
dtype='object')
```

```
[67]: for col in train_d.columns:
print(col,)
```

```
LotShape
LotConfig
HouseStyle
Exterior1st
```

ExterQual
Foundation
BsmtQual
BsmtExposure
HeatingQC
KitchenQual
FireplaceQu
OverallQual
GrLivArea
ExterQual
KitchenQual
TotalBsmtSF
GarageCars
1stFlrSF
GarageArea
BsmtQual
FullBath
TotRmsAbvGrd
FireplaceQu
Foundation
BsmtFinSF2
MiscVal
Id
OverallCond
EnclosedPorch
SalePrice
Neighborhood_1
Neighborhood_2
Neighborhood_3
Neighborhood_4
Neighborhood_5
Neighborhood_6
Neighborhood_7
Neighborhood_8
Neighborhood_9
Neighborhood_10
Neighborhood_mes
GarageType_2Types
GarageType_Attchd
GarageType_Basment
GarageType_BuiltIn
GarageType_CarPort
GarageType_Detachd
GarageType_None
GarageFinish_Fin
GarageFinish_None
GarageFinish_RFn
GarageFinish_Unf

Neighborhood_1
Neighborhood_2
Neighborhood_3
Neighborhood_4
Neighborhood_5
Neighborhood_6
Neighborhood_7
Neighborhood_8
Neighborhood_9
Neighborhood_10
Neighborhood_mes
GarageFinish_Fin
GarageFinish_None
GarageFinish_RFn
GarageFinish_Unf

```
[68]: from sklearn.linear_model import LinearRegression
      from sklearn.model_selection import train_test_split
      predictors=train_d.drop(['Id', 'SalePrice'],axis=1)
      target=train.SalePrice
      x_train,x_cv,y_train,y_cv=train_test_split(predictors,target,test_size=0.
      ↪35,random_state=0)
```

```
[69]: from sklearn.model_selection import KFold,cross_val_score
      logreg = LinearRegression()
      logreg.fit(x_train, y_train)
```

[69]: LinearRegression()

```
[70]: logreg.predict(x_cv)
```

```
[70]: array([144139.43782158, 152744.003098 , 43963.37022548, 127465.94601343,
          269474.27559947, 120459.03952188, 208047.63481301, 117075.56596074,
          139363.55870788, 141787.84145595, 200122.06941272, 101730.18720464,
          109021.79338203, 151590.06313489, 100603.60401613, 247132.86577 ,
          217246.115668 , 79123.65915782, 165009.0824828 , 389122.19903686,
          303928.97626247, 106442.9142123 , 121999.08365686, 107320.6988759 ,
          105894.79780088, 225869.4817535 , 201594.31318564, 100798.1195574 ,
          140659.93627001, 396144.61414194, 174525.60224797, 221849.04495508,
          144402.43231512, 240588.32413429, 87665.39512821, 113179.88204427,
          244369.69716486, 280094.39038557, 94168.41635413, 68620.57308695,
          129230.2181316 , 246004.26136834, 127420.15010849, 117961.18099309,
          184893.35462445, 176174.65556824, 315266.00303225, 327251.94531695,
          422163.98599134, 261506.8430623 , 247736.76714408, 98545.3165135 ,
          151413.99421624, 119430.74990179, 404669.22144335, 252180.81074967,
          307280.356098 , 83519.49957952, 195376.21498372, 126644.05136689,
          178446.22793128, 125911.58601471, 125170.75335233, 267354.49417125,
```

105224.31337158, 360315.17626491, 286088.62270053, 263160.36407324,
152749.05521216, 97724.01931367, 124427.32145368, 241221.38862815,
87763.73267459, 156830.70755337, 382548.8476037 , 157917.09363142,
248035.79653976, 226585.44026249, 247006.78802771, 227329.84475522,
213483.28690774, 129677.14535453, 268924.79587656, 228703.04988093,
210169.65185134, 140090.01812955, 135344.0279052 , 139194.34722826,
147080.38203471, 95399.2430098 , 287521.86415949, 143295.71991319,
128640.3272776 , 290851.08193852, 356965.4888421 , 94336.38001244,
302092.45999088, 141539.097783 , 182675.63491664, 102620.57137709,
209157.36313489, 206720.44722455, 210044.29366863, 105426.12995542,
184206.20342047, 183301.53431923, 218433.05806976, 211397.07602481,
159450.9993818 , 251142.18340008, 115124.53397807, 148950.20292673,
225346.74219983, 120940.04728361, 111798.02013579, 111509.47292834,
328973.03909941, 238869.26129253, 317895.24724495, 138184.71694764,
143584.19280668, 78939.51903769, 77749.39264872, 27908.27440554,
11193.45148136, 190728.22085807, 109947.64923276, 279975.70485738,
244522.09966044, 176860.15918987, 328643.93738724, 143440.93436935,
205517.40942247, 267440.0420748 , 238517.09642814, 196145.68987838,
139759.10098191, 147739.59612601, 112884.54818032, 114639.93542004,
312736.84205578, 234577.43423057, 168040.27076859, 107034.59875851,
139528.83162636, 117249.3256618 , 209866.83236219, 254738.71011224,
157874.38494365, 159203.81369684, 78109.16306149, 202827.63726737,
276747.3452069 , 157097.88992072, 273858.53147888, 93426.9945409 ,
181704.68294939, 112115.40169026, 424264.86061613, 129068.80055719,
76932.25063546, 188130.74484756, 429807.87297376, 159001.80153041,
179358.4469029 , 123871.98737571, 237675.38255669, 105361.97715064,
98811.85871755, 151559.71220274, 119266.33678231, 177469.52975423,
273884.0038697 , 288184.90396019, 387309.41996646, 143030.83376884,
154914.01835932, 163774.74449677, 95966.8882176 , 101263.28028888,
136002.59606922, 275030.51498242, 239887.88825755, 131994.49554582,
301700.15809705, 187734.40038137, 173152.21628308, 181906.61010418,
147892.65762258, 138172.35881784, 174774.98226032, 339749.5536299 ,
188409.01971294, 70516.04796421, 222875.73645817, 209924.59358787,
204951.54199666, 115750.20439879, 219974.14847263, 141389.55730856,
118736.9518591 , 227059.18152322, 200224.93571953, 335657.7343163 ,
228017.7927083 , 219798.31304757, 184433.50289457, 259097.72138749,
244766.70575466, 210250.46835843, 183870.43034126, 145542.44950262,
231445.99076327, 281811.41362833, 288108.28914281, 99287.69244015,
317962.80333296, 241329.23111765, 193892.65524767, 124520.01446085,
87695.38860199, 221466.74883663, 247591.29767046, 307994.180637 ,
242588.296742 , 205637.12519361, 37387.29575294, 191003.6951427 ,
107800.30708232, 267949.67041092, 209521.03621616, 223106.61465036,
206099.99211183, 238536.85596912, 218512.1119728 , 166759.79461476,
147767.68498584, 228470.31404564, 179021.32254741, 165823.05871153,
249189.95977599, 395138.38432891, 133286.85624476, 127607.27558259,
122817.61302668, 185266.45034532, 150996.06930924, 320058.00931214,
242715.53276902, 102110.41657847, 223959.99570204, 127803.40218124,

105477.66492271, 359091.45157812, 88104.27704704, 237608.55741504,
 214417.62918947, 209163.57318925, 201024.4411978 , 128920.96670989,
 35497.25969094, 170574.50812411, 249344.00063712, 271459.62548362,
 386614.69461994, 134169.28568877, 181698.72815915, 177008.7344555 ,
 64290.29285191, 191544.89033911, 147958.50008738, 168334.77940608,
 224150.75458841, 96688.60572267, 61195.01254213, 156124.35924177,
 153871.92340607, 184190.82182548, 161157.61737204, 198918.5487023 ,
 195822.46091625, 78736.74459925, 244757.61147235, 153633.61406739,
 75966.46534818, 87231.35154395, 169108.6440893 , 134233.8243936 ,
 173876.24829434, 147899.91216084, 94096.79817591, 144949.11382236,
 156211.08041319, 95384.17385805, 123132.49698952, 165156.892182 ,
 271877.49517966, 345850.93571236, 167893.15966202, 112849.1249278 ,
 93702.7963833 , 309409.22247562, 180808.1788399 , 92363.20145638,
 297552.92095788, 151233.04665853, 189534.8304892 , 121848.9427948 ,
 193493.66585478, 275760.94462485, 234646.97942894, 217819.68584261,
 74986.66414188, 132161.63495456, 175861.67370333, 159327.16772935,
 156107.81524039, 146810.1182511 , 54999.97434404, 246086.37541331,
 222702.92738073, 121892.42003042, 140532.95917254, 181723.66960687,
 204849.43994863, 319827.67821991, 165635.82758451, 134288.45839938,
 240127.05877187, 449470.85926137, 278708.3544191 , 139989.28208357,
 135006.84218006, 144034.95551285, 275133.79236415, 123763.27751326,
 107593.80871095, 208604.42308566, 137417.5602695 , 182761.5622265 ,
 127648.16935887, 117456.99181404, 207783.90607221, 206565.33151113,
 347961.30652877, 49469.42051405, 49219.82563653, 160570.89741594,
 50681.78349739, 306877.75485814, 240020.76602238, 267416.53446327,
 277763.62017803, 235727.48328516, 330065.74410962, 213736.30273349,
 255607.27724153, 131721.74187174, 111724.80103649, 183339.12131031,
 66632.79518138, 45985.01081933, 61019.12244289, 102795.26612559,
 127472.66156331, 141064.49339932, 99945.00276147, 118447.23790914,
 86582.55759194, 96737.77548498, 271316.23659865, 127429.45252153,
 246287.25444199, 148945.62216476, 318349.12472756, 140878.80274351,
 125007.15348156, 119166.88173825, 104967.86530836, 156195.89670746,
 136152.83060205, 117850.75708343, 103210.14308626, 170809.63743224,
 98861.29276134, 126165.55459338, 279929.28590297, 121111.88986101,
 138870.79746572, 356600.76628159, 178623.64800015, 165278.4446023 ,
 101950.98975152, 146113.04117439, 222528.16239288, 107463.95320323,
 140119.82258214, 148090.74801253, 161186.86733772, 137705.20301748,
 189544.36704689, 252617.21843019, 87428.24032829, 136838.59000293,
 210996.65366663, 142190.5668071 , 149011.92832306, 189614.82497152,
 217574.14186913, 118608.95420924, 179084.51672122, 142435.63331699,
 358251.07480691, 324329.82600105, 251078.86712584, 157712.89221188,
 155951.75533057, 106896.20017626, 291110.50599234, 159534.21961308,
 45819.54258843, 114704.79187328, 107981.57508237, 108475.63341791,
 195598.74094284, 214956.35800981, 252815.25825903, 154722.89687038,
 74245.24997319, 130128.42754591, 213521.17995387, 214885.34158937,
 129087.828529 , 93937.91279295, 184701.94885444, 154186.69474288,
 270208.02966664, 192917.63047559, 118230.79872431, 45511.61520124,

```

175962.25748192, 232652.53336739, 117406.54485017, 111211.14043969,
247097.60095826, 136862.51941046, 160888.41626066, 143995.51017993,
306531.64396896, 89365.34918077, 110382.9537918 , 94808.28954602,
253829.20232269, 307008.9604607 , 110745.02209559, 127811.90016872,
271996.42150266, 284259.82684483, 160192.39862678, 140304.965622 ,
143321.00676402, 239880.19428086, 120467.08104153, 125137.36095609,
107428.56243938, 212125.0990565 , 126502.96239181, 178405.52146671,
101281.68405047, 217374.16470075, 249749.73363216, 104089.88531109,
183476.17357519, 136489.51422201, 195222.01296852, 363661.44117254,
243569.6022992 , 147310.81157537, 125304.23075252, 110039.59504011,
87558.96197101, 94535.25625197, 116928.61377453, 196287.77876077,
184671.06624629, 157963.99051315, 159508.30687849, 260378.2212877 ,
226171.02616223, 159998.30791641, 125903.41620229, 226086.02344035,
113677.6658485 , 145187.63746163, 110409.79490066, 263610.61013119,
157545.25945803, 346067.10181446, 237553.69743181, 145828.9881826 ,
137905.97212915, 162005.31125641, 396941.17652426, 124631.21994582,
230304.13844323, 109919.00432701, 122962.48218907, 109334.17498207,
179532.40511388, 178545.52231007, 134506.31471075])

```

```

[71]: from sklearn.metrics import accuracy_score
      y_pred=logreg.predict(x_cv)
      logreg.score(x_cv,y_cv)

```

[71]: 0.8637143770467263

```

[72]: features_train=['LotShape', 'LotConfig', 'Neighborhood', 'HouseStyle',
↳ 'Exterior1st',
      'ExterQual', 'Foundation', 'BsmtQual', 'BsmtExposure', 'HeatingQC',
      'KitchenQual', 'FireplaceQu', 'GarageType', 'GarageFinish',
      'OverallQual', 'GrLivArea', 'Neighborhood', 'ExterQual', 'KitchenQual',
      'TotalBsmtSF', 'GarageCars', '1stFlrSF', 'GarageArea', 'BsmtQual',
      'FullBath', 'GarageFinish', 'TotRmsAbvGrd', 'FireplaceQu', 'Foundation',
      'BsmtFinSF2', 'MiscVal', 'OverallCond', 'EnclosedPorch']

```

[]: