In [156]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from scipy.stats import norm
from sklearn.preprocessing import StandardScaler
from scipy import stats
import warnings
warnings.filterwarnings('ignore')

%matplotlib inline
```

In [2]:

```
df_train = pd.read_csv('train.csv')
df_train.head()
```

Out[2]:

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

5 rows × 81 columns

In [3]:

```
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
Id               1460 non-null int64
MSSubClass       1460 non-null int64
MSZoning         1460 non-null object
LotFrontage      1201 non-null float64
LotArea          1460 non-null int64
Street           1460 non-null object
Alley            91 non-null object
LotShape         1460 non-null object
LandContour      1460 non-null object
Utilities        1460 non-null object
LotConfig        1460 non-null object
LandSlope        1460 non-null object
Neighborhood     1460 non-null object
Condition1       1460 non-null object
Condition2       1460 non-null object
BldgType         1460 non-null object
HouseStyle       1460 non-null object
OverallQual      1460 non-null int64
OverallCond      1460 non-null int64
YearBuilt        1460 non-null int64
YearRemodAdd     1460 non-null int64
RoofStyle        1460 non-null object
RoofMatl         1460 non-null object
Exterior1st      1460 non-null object
Exterior2nd      1460 non-null object
MasVnrType       1452 non-null object
MasVnrArea       1452 non-null float64
ExterQual        1460 non-null object
ExterCond        1460 non-null object
Foundation       1460 non-null object
BsmtQual         1423 non-null object
BsmtCond         1423 non-null object
BsmtExposure     1422 non-null object
BsmtFinType1     1423 non-null object
BsmtFinSF1       1460 non-null int64
BsmtFinType2     1422 non-null object
BsmtFinSF2       1460 non-null int64
BsmtUnfSF        1460 non-null int64
TotalBsmtSF      1460 non-null int64
Heating          1460 non-null object
HeatingQC        1460 non-null object
CentralAir       1460 non-null object
Electrical       1459 non-null object
1stFlrSF         1460 non-null int64
2ndFlrSF         1460 non-null int64
LowQualFinSF     1460 non-null int64
GrLivArea        1460 non-null int64
BsmtFullBath     1460 non-null int64
BsmtHalfBath     1460 non-null int64
FullBath         1460 non-null int64
HalfBath         1460 non-null int64
BedroomAbvGr     1460 non-null int64
KitchenAbvGr     1460 non-null int64
KitchenQual      1460 non-null object
```

```
TotRmsAbvGrd     1460 non-null int64
Functional       1460 non-null object
Fireplaces       1460 non-null int64
FireplaceQu      770 non-null object
GarageType       1379 non-null object
GarageYrBlt      1379 non-null float64
GarageFinish     1379 non-null object
GarageCars       1460 non-null int64
GarageArea       1460 non-null int64
GarageQual       1379 non-null object
GarageCond       1379 non-null object
PavedDrive       1460 non-null object
WoodDeckSF       1460 non-null int64
OpenPorchSF      1460 non-null int64
EnclosedPorch    1460 non-null int64
3SsnPorch        1460 non-null int64
ScreenPorch      1460 non-null int64
PoolArea         1460 non-null int64
PoolQC           7 non-null object
Fence            281 non-null object
MiscFeature      54 non-null object
MiscVal          1460 non-null int64
MoSold           1460 non-null int64
YrSold           1460 non-null int64
SaleType         1460 non-null object
SaleCondition    1460 non-null object
SalePrice        1460 non-null int64
dtypes: float64(3), int64(35), object(43)
memory usage: 924.0+ KB
```

In [4]:

```
df_train.describe(include = 'all')
```

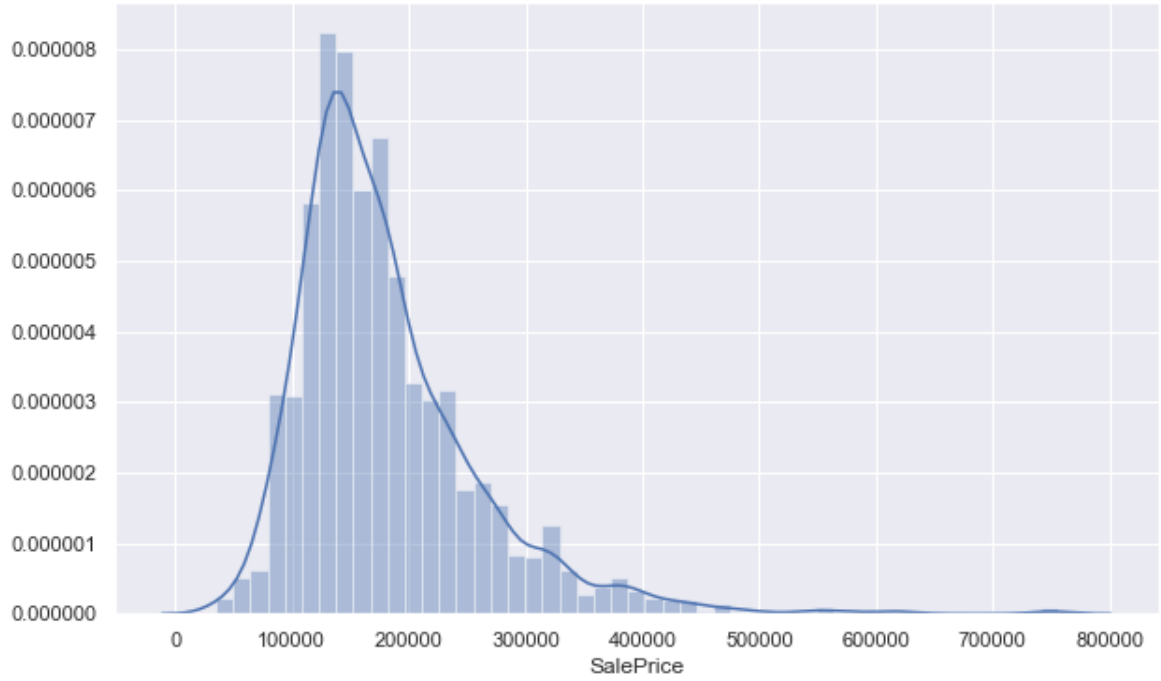
Out[4]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotSh
count	1460.000000	1460.000000	1460	1201.000000	1460.000000	1460	91	1
unique	NaN	NaN	5	NaN	NaN	2	2	
top	NaN	NaN	RL	NaN	NaN	Pave	Grvl	I
freq	NaN	NaN	1151	NaN	NaN	1454	50	
mean	730.500000	56.897260	NaN	70.049958	10516.828082	NaN	NaN	N
std	421.610009	42.300571	NaN	24.284752	9981.264932	NaN	NaN	N
min	1.000000	20.000000	NaN	21.000000	1300.000000	NaN	NaN	N
25%	365.750000	20.000000	NaN	59.000000	7553.500000	NaN	NaN	N
50%	730.500000	50.000000	NaN	69.000000	9478.500000	NaN	NaN	N
75%	1095.250000	70.000000	NaN	80.000000	11601.500000	NaN	NaN	N
max	1460.000000	190.000000	NaN	313.000000	215245.000000	NaN	NaN	N

11 rows × 81 columns

In [5]:

```
plt.figure(figsize=(10,6))
sns.set()
sns.distplot(df_train['SalePrice'])
plt.show()
```



In [6]:

```
f,axis = plt.subplots(figsize = (10,6))
data = pd.concat([df_train['GrLivArea'],df_train['SalePrice']],axis = 1)
f = data.plot.scatter(y = 'SalePrice',x = 'GrLivArea',ax =axis)

plt.show()
```

'c' argument looks like a single numeric RGB or RGBA sequence, which should be avoid
ed as value-mapping will have precedence in case its length matches with 'x' & 'y'.
Please use a 2-D array with a single row if you really want to specify the same RGB
or RGBA value for all points.

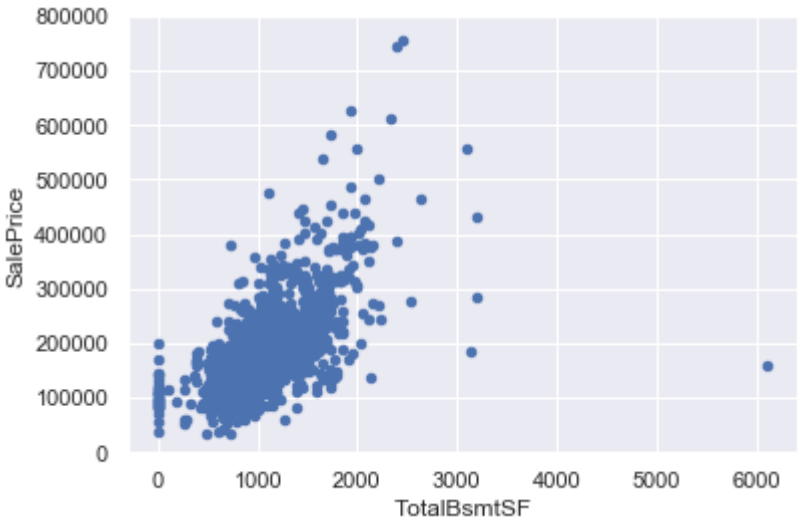
In [7]:

```
data = pd.concat([df_train['SalePrice'], df_train['TotalBsmtSF']], axis=1)
data.plot.scatter(x="TotalBsmtSF", y='SalePrice', ylim=(0,800000))
```

'c' argument looks like a single numeric RGB or RGBA sequence, which should be avoid
ed as value-mapping will have precedence in case its length matches with 'x' & 'y'.
Please use a 2-D array with a single row if you really want to specify the same RGB
or RGBA value for all points.

Out[7]:

<matplotlib.axes._subplots.AxesSubplot at 0xafe62b0>

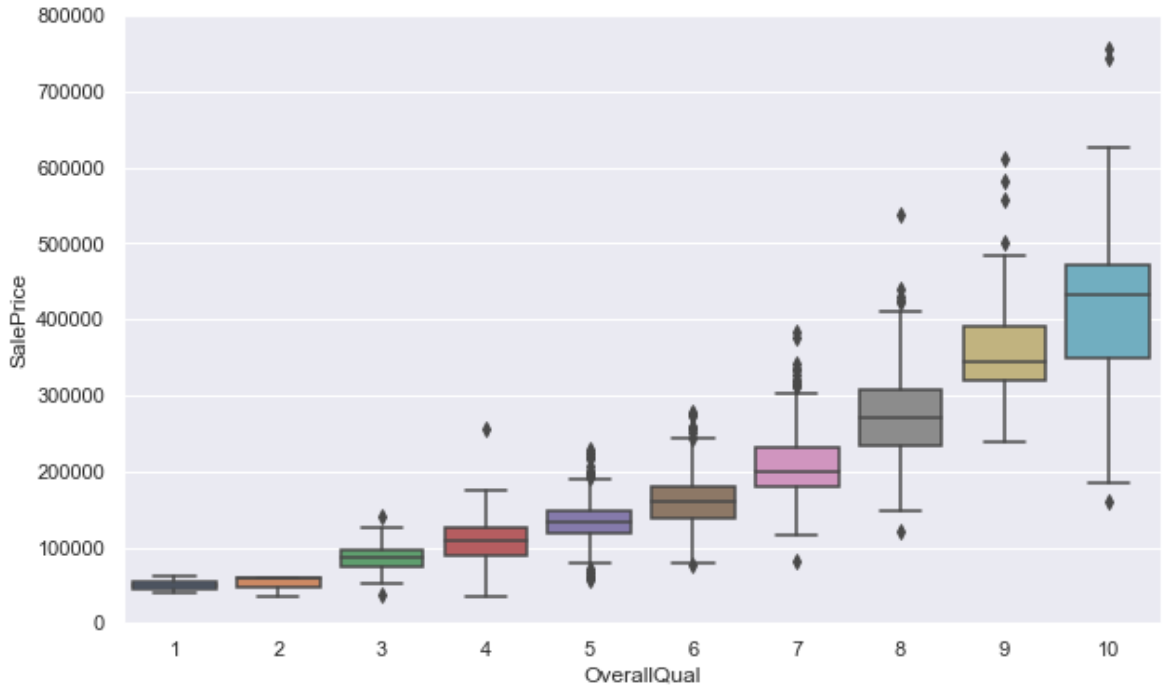


In [8]:

```
fig ,qaxis = plt.subplots(figsize =(10,6))
data = pd.concat([df_train['SalePrice'], df_train['OverallQual']], axis=1)
fig = sns.boxplot(data = data,x = 'OverallQual',y = 'SalePrice')
qaxis.set(ylim=(0,800000))
```

Out[8]:

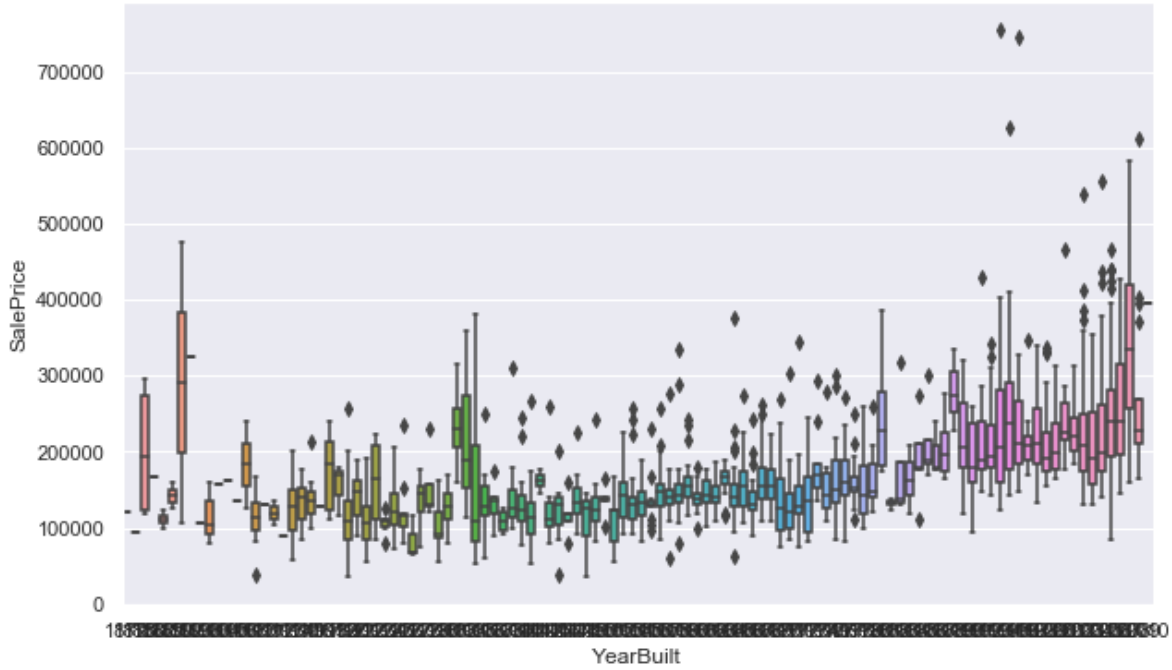
```
[(0, 800000)]
```



In [9]:

```
a = plt.figure(figsize = (10,6))

data = pd.concat([df_train['YearBuilt'],df_train['SalePrice']],axis = 1)
sns.boxplot(data = data ,x = 'YearBuilt',y = 'SalePrice')
plt.show()
```

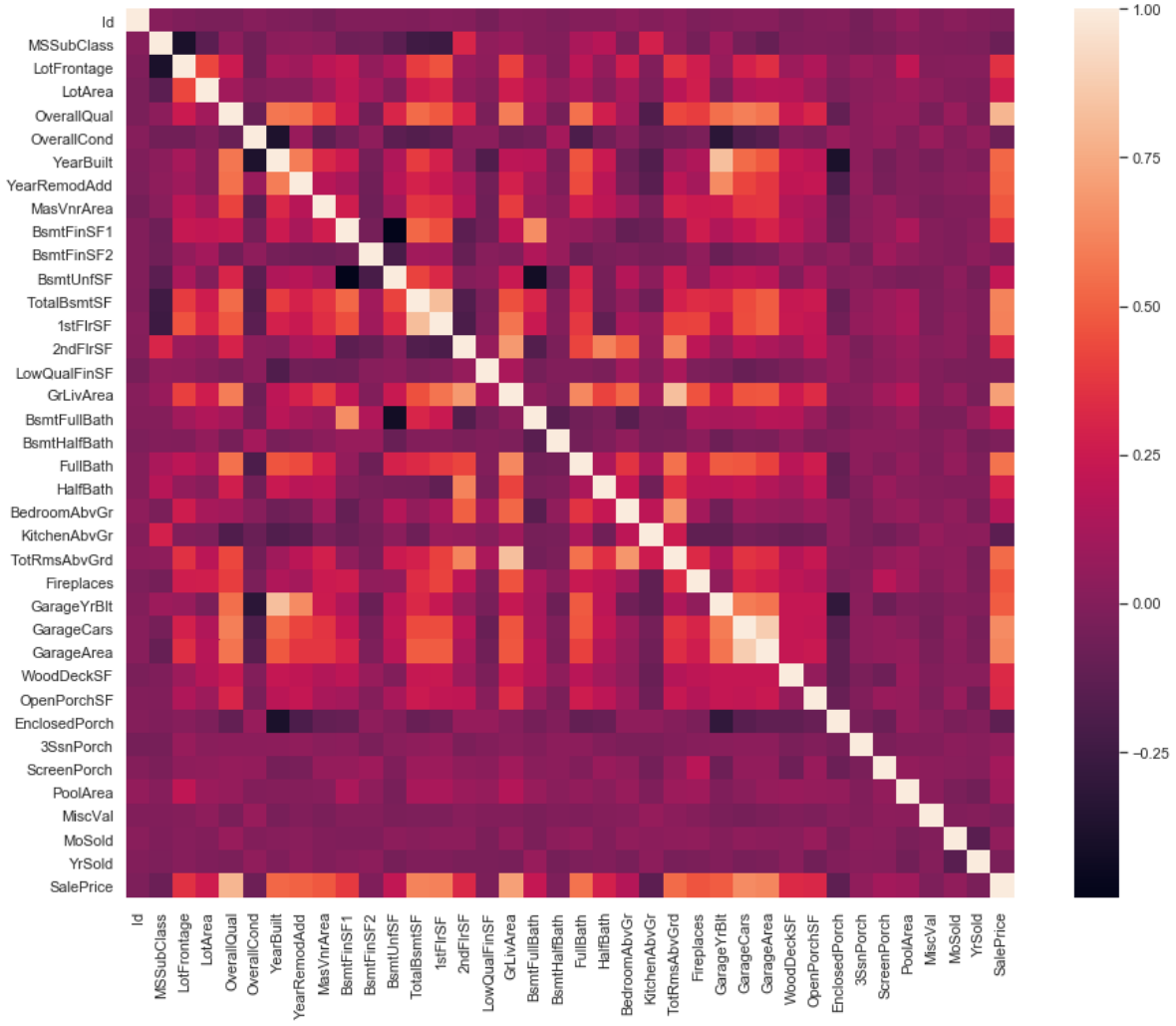


In [10]:

```
plt.figure(figsize = (16,12))
sns.heatmap(df_train.corr(),square = True)
```

Out[10]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xb893048>
```



In [13]:

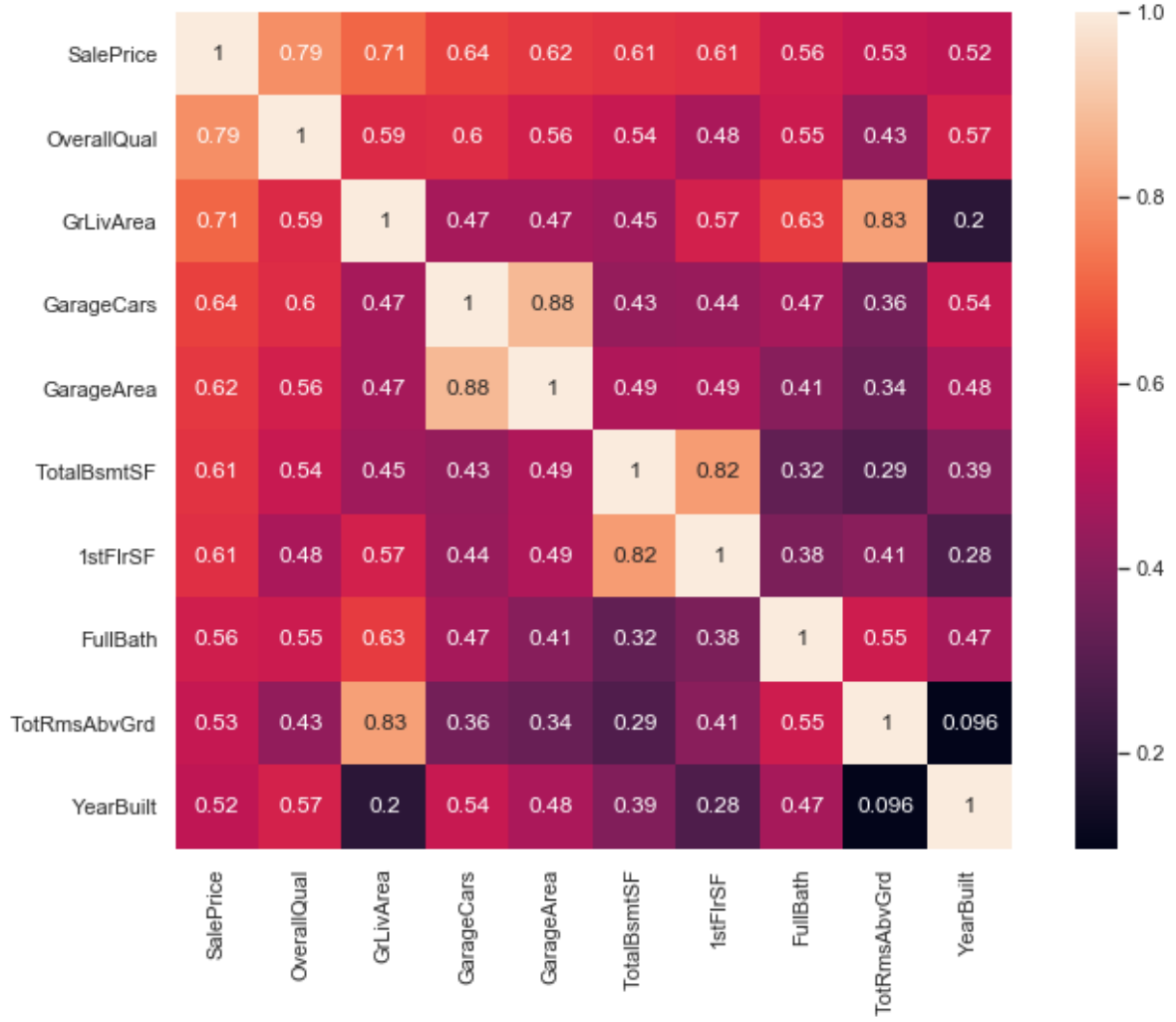
```
val = df_train.corr()
col = val.nlargest(10,'SalePrice').index
col
```

Out[13]:

```
Index(['SalePrice', 'OverallQual', 'GrLivArea', 'GarageCars', 'GarageArea',
       'TotalBsmtSF', '1stFlrSF', 'FullBath', 'TotRmsAbvGrd', 'YearBuilt'],
      dtype='object')
```

In [32]:

```
data_heat = pd.DataFrame(np.corrcoef(df_train[col].values.T), index = col, columns = col)
plt.figure(figsize = (12,8))
sns.heatmap(data_heat, square = True, annot = True)
plt.show()
```



In [33]:

```
df_train.head()
```

Out[33]:

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

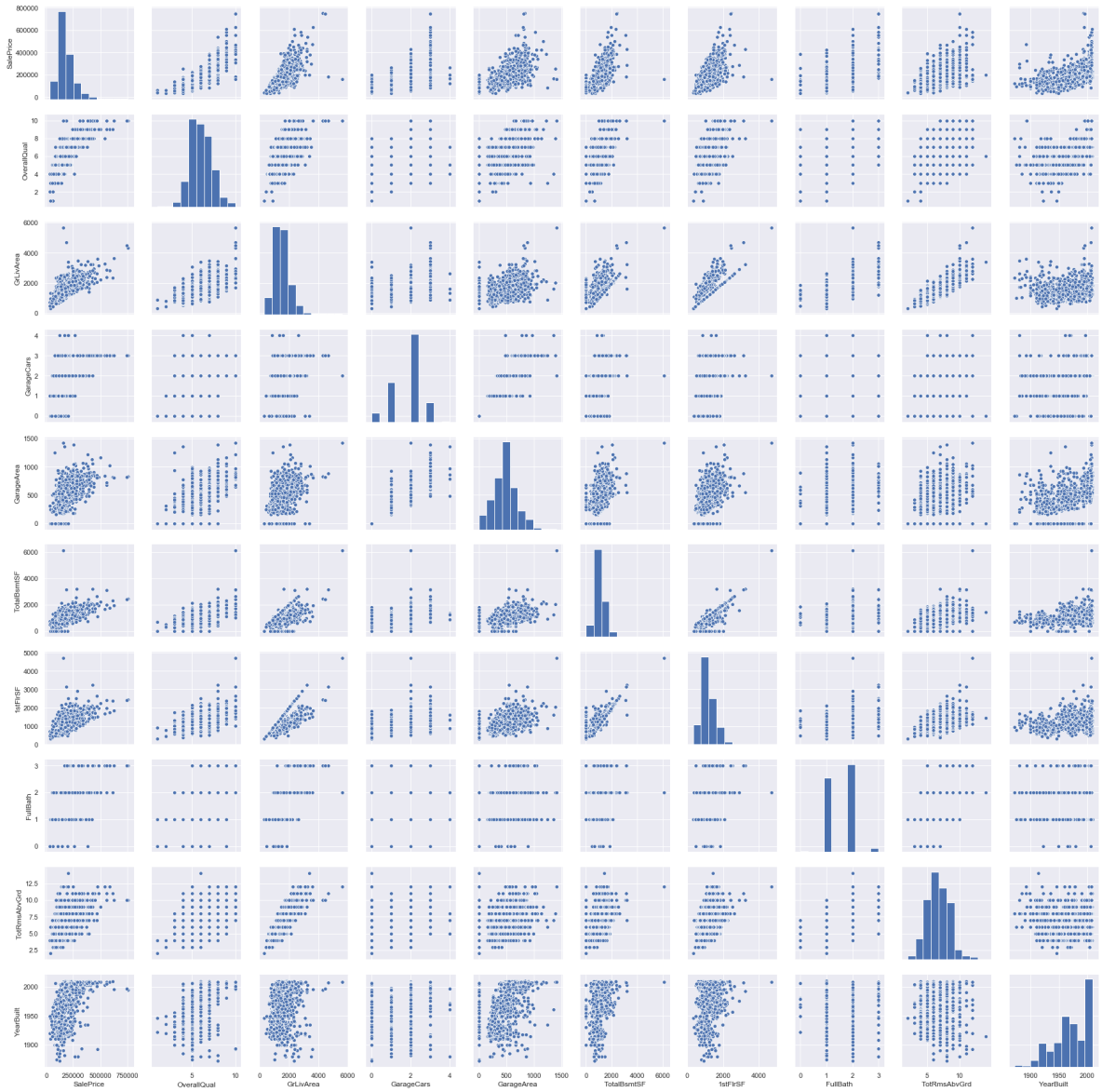
5 rows × 81 columns

In [36]:

```
sns.pairplot(df_train[col])
```

Out[36]:

<seaborn.axisgrid.PairGrid at 0x13d511d0>



In [51]:

```
miss_acount = df_train.isnull().sum().sort_values(ascending = False)
miss_rate = missing_acount/df_train.shape[0]
miss_overview = pd.concat([acount,rate],axis = 1,keys = ['acount','rate'])
miss_overview.head(20)
```

Out[51]:

	acount	rate
PoolQC	1453	0.995205
MiscFeature	1406	0.963014
Alley	1369	0.937671
Fence	1179	0.807534
FireplaceQu	690	0.472603
LotFrontage	259	0.177397
GarageCond	81	0.055479
GarageType	81	0.055479
GarageYrBlt	81	0.055479
GarageFinish	81	0.055479
GarageQual	81	0.055479
BsmtExposure	38	0.026027
BsmtFinType2	38	0.026027
BsmtFinType1	37	0.025342
BsmtCond	37	0.025342
BsmtQual	37	0.025342
MasVnrArea	8	0.005479
MasVnrType	8	0.005479
Electrical	1	0.000685
Utilities	0	0.000000

In [55]:

```
df_train.drop(miss_overview[miss_overview.acount>1].index,axis = 1,inplace = True)
```

In [59]:

```
df_train.isnull().sum().sort_values(ascending = False).head()
```

Out[59]:

```
Electrical    1
SalePrice     0
Heating       0
BsmtUnfSF     0
BsmtFinSF2    0
dtype: int64
```

In [34]:

```
df_train['TotalBsmtSF'].value_counts().sort_index(ascending = True)[:1]
```

Out[34]:

```
0    37
Name: TotalBsmtSF, dtype: int64
```

In [38]:

```
df_train.head()
```

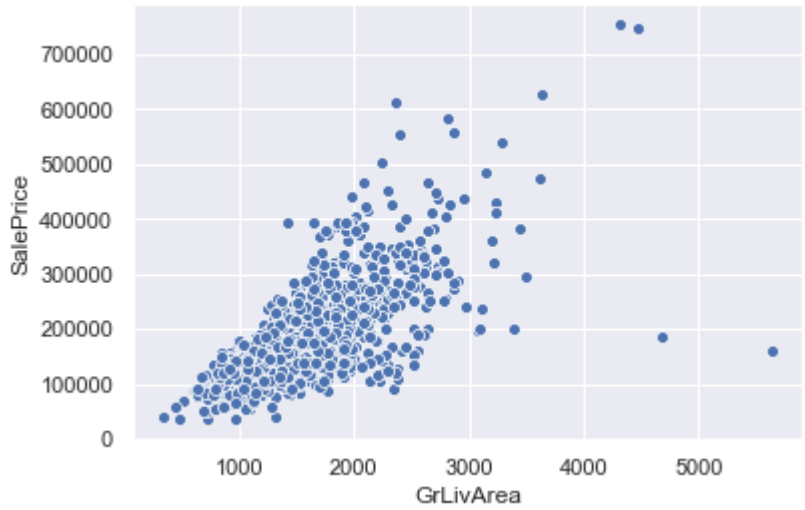
Out[38]:

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

5 rows × 81 columns

In [69]:

```
data = pd.concat([df_train['GrLivArea'],df_train['SalePrice']],axis = 1)
sns.scatterplot(data = data,x = 'GrLivArea',y = 'SalePrice')
plt.show()
```



In [75]:

```
data.sort_values(by = 'GrLivArea',ascending = False).head()
```

Out[75]:

	GrLivArea	SalePrice
1298	5642	160000
523	4676	184750
1182	4476	745000
691	4316	755000
1169	3627	625000

In [78]:

```
df_train.loc[1298][['GrLivArea','SalePrice']]
```

Out[78]:

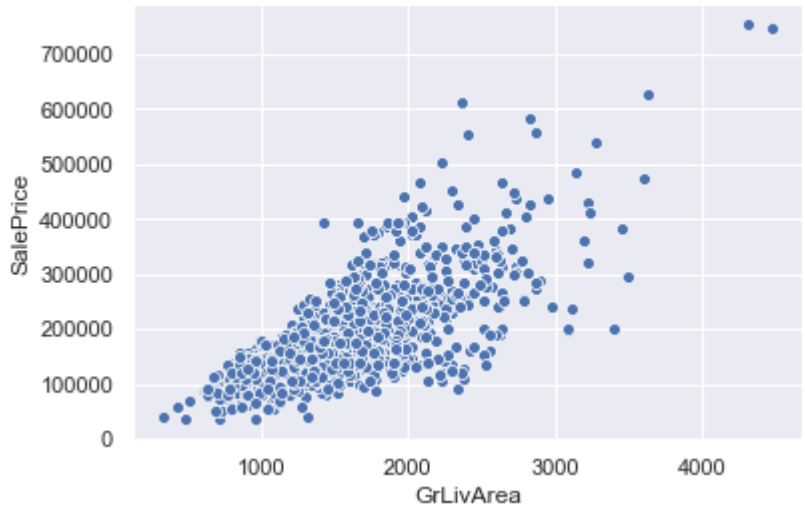
```
GrLivArea      5642
SalePrice    160000
Name: 1298, dtype: object
```

In [113]:

```
df_train.drop([1298,523],inplace = True)
```

In [114]:

```
data = pd.concat([df_train['GrLivArea'],df_train['SalePrice']],axis = 1)
sns.scatterplot(data = data,x = 'GrLivArea',y = 'SalePrice')
plt.show()
```



In [119]:

```
data = pd.concat([df_train['GrLivArea'],df_train['SalePrice']],axis = 1)
sns.scatterplot(data = data,x = 'SalePrice',y = 'GrLivArea')
```

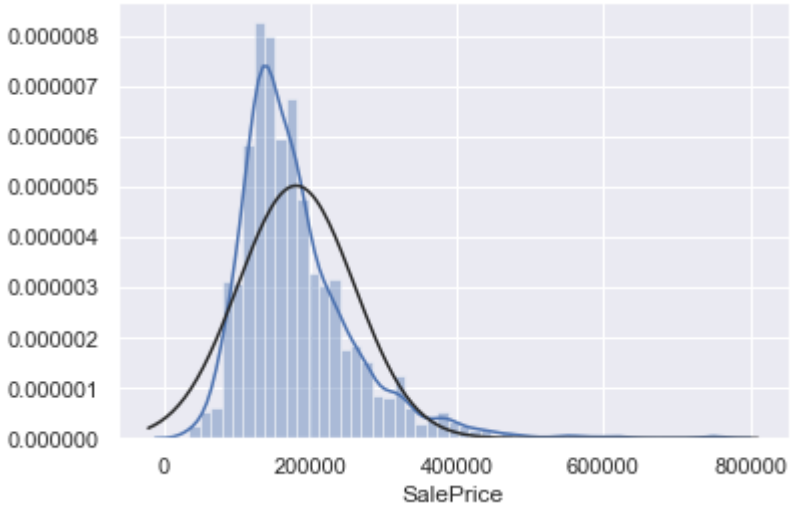
Out[119]:

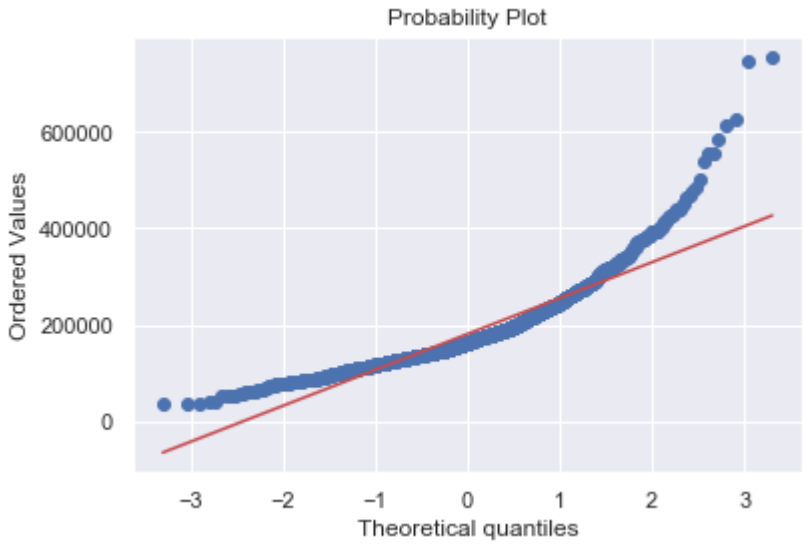
<matplotlib.axes._subplots.AxesSubplot at 0x194d01d0>

In [172]:

```
a = sns.distplot(df_train['SalePrice'],fit = norm)

plt.figure()
b = stats.probplot(df_train['SalePrice'],plot = plt)
```



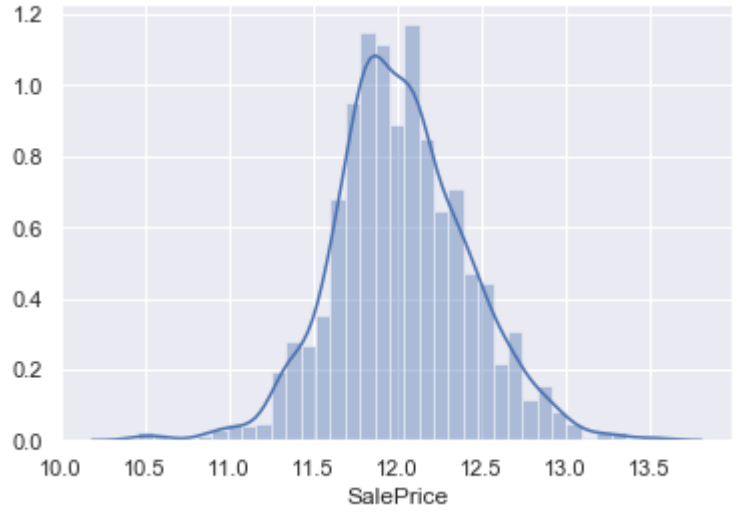


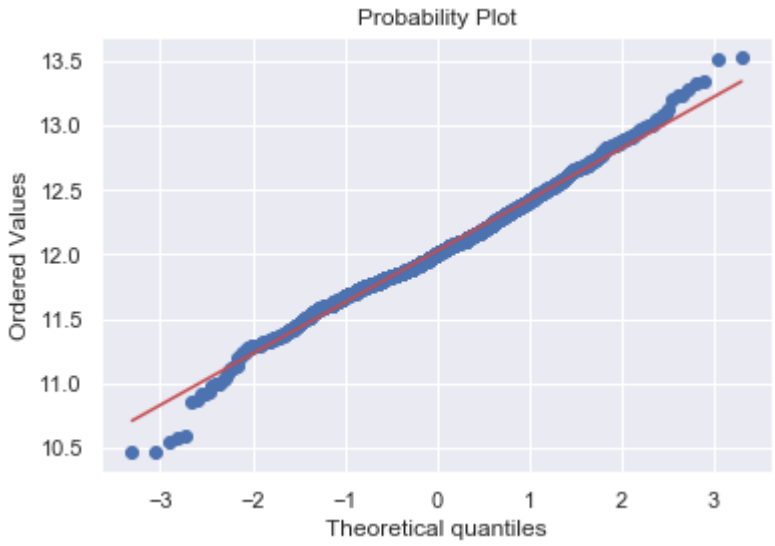
In [180]:

```
_ = sns.distplot(np.log(df_train['SalePrice']))

plt.figure()

_ = stats.probplot(np.log(df_train['SalePrice']),plot = plt)
```



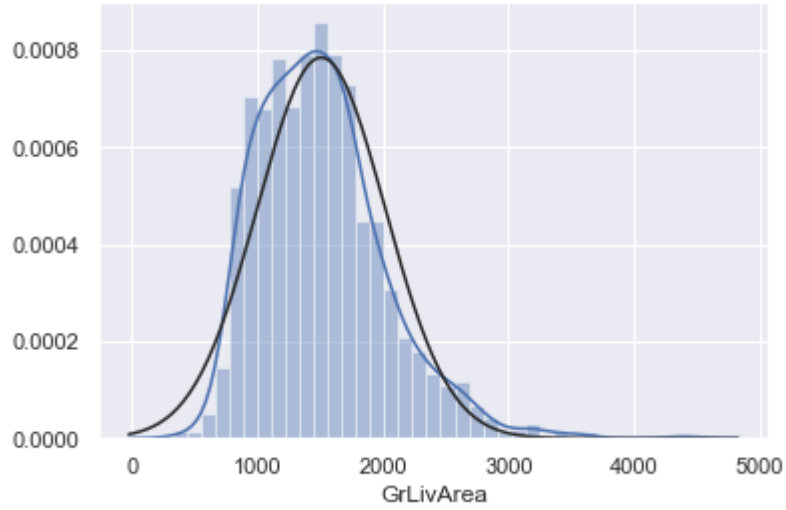


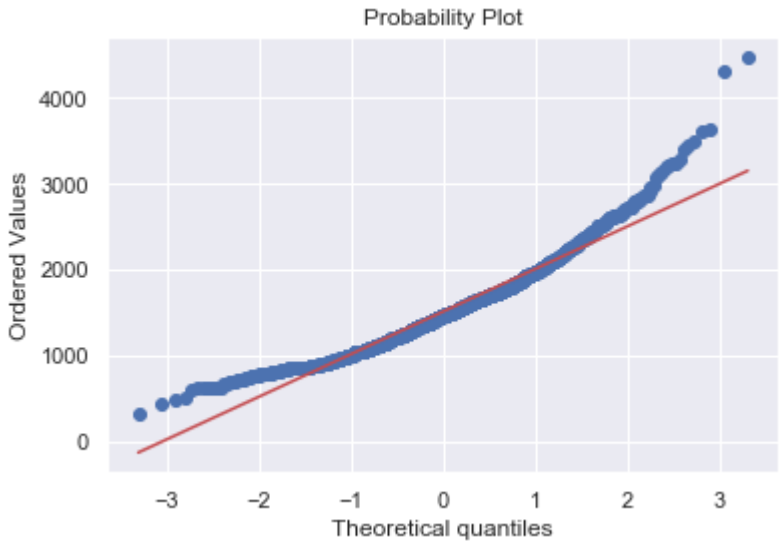
In [184]:

```
_ = sns.distplot(df_train['GrLivArea'],fit = norm)

plt.figure()

_ = stats.probplot(df_train['GrLivArea'],plot = plt)
```



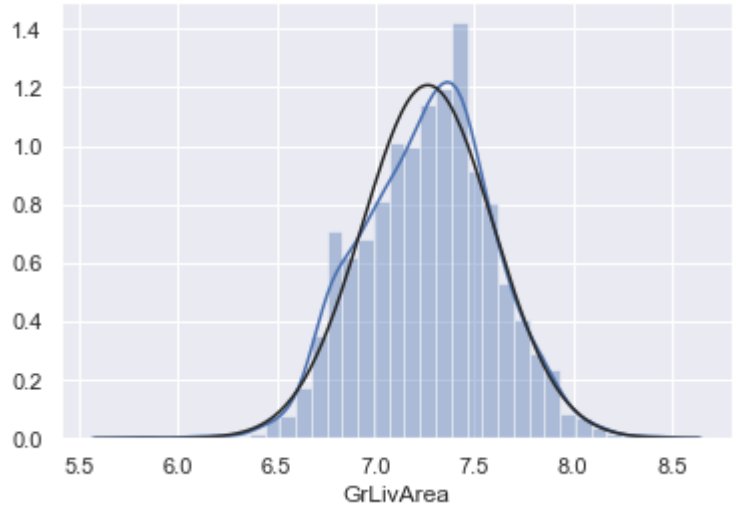


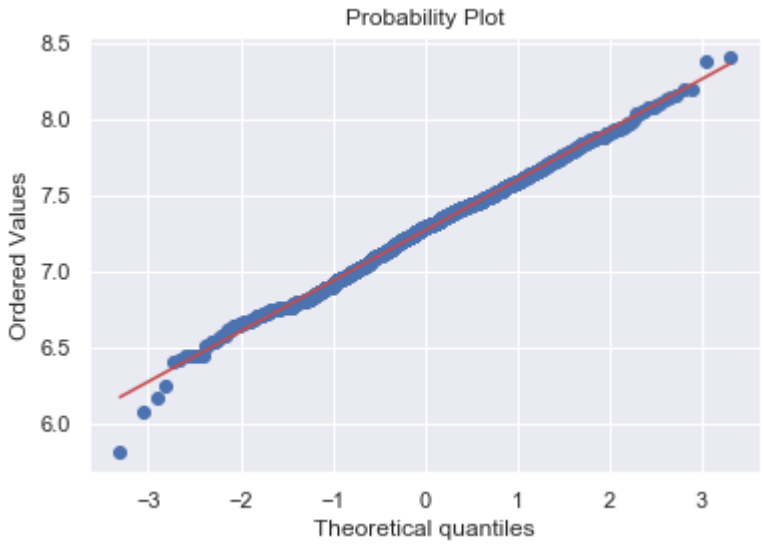
In [188]:

```
_ = sns.distplot(np.log(df_train['GrLivArea']),fit = norm)

plt.figure()

_ = stats.probplot(np.log(df_train['GrLivArea']),plot = plt)
```



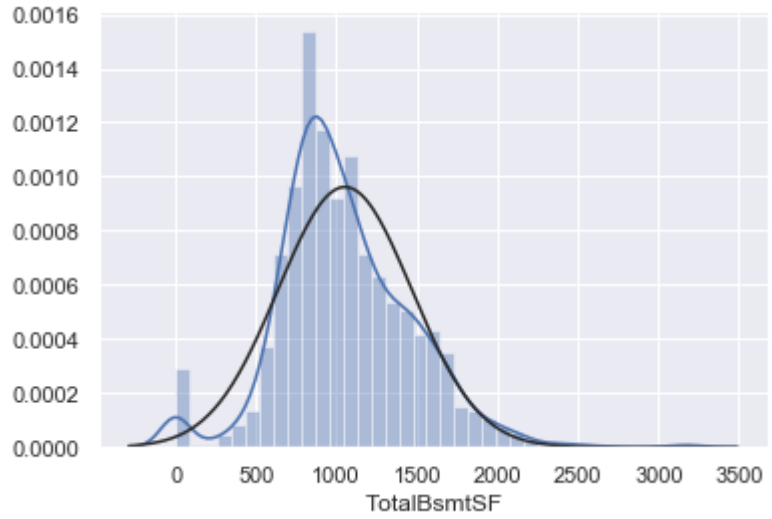


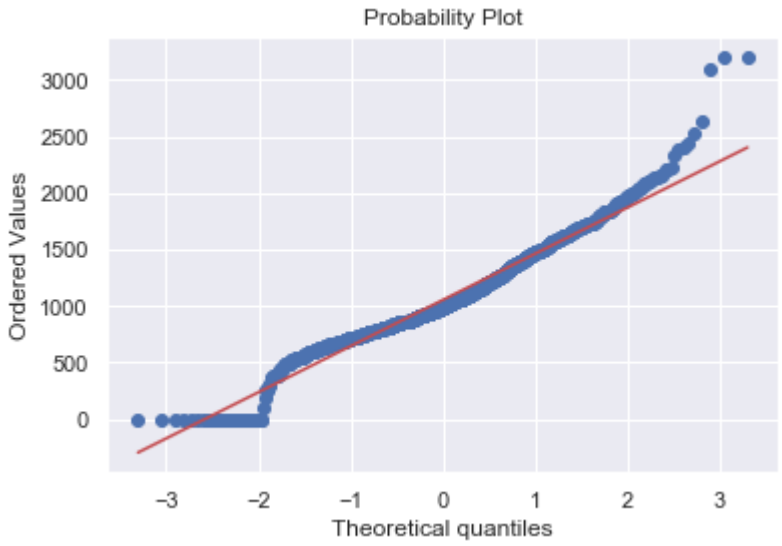
In [193]:

```
_ = sns.distplot(df_train['TotalBsmtSF'],fit = norm)

plt.figure()

_ = stats.probplot(df_train['TotalBsmtSF'],plot = plt)
```

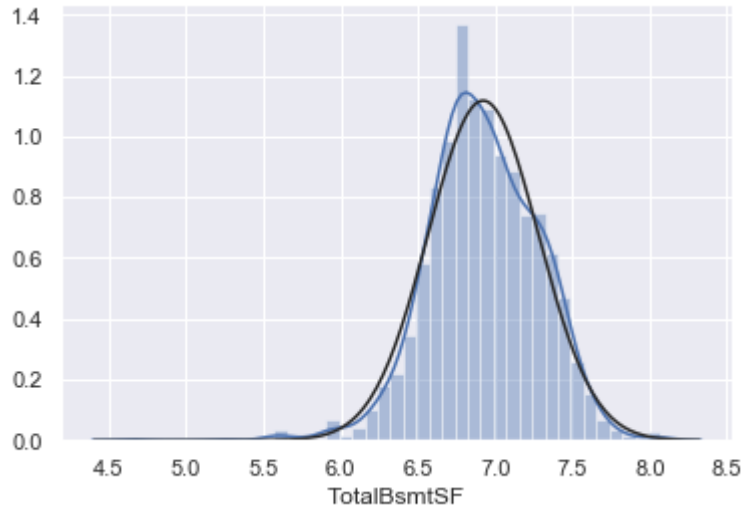


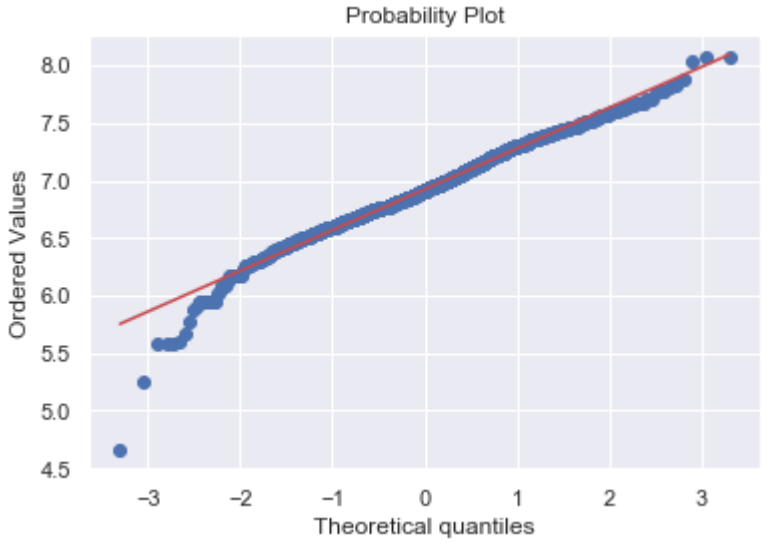


In [200]:

```
_  = sns.distplot(np.log(df_train[df_train['TotalBsmtSF']!=0]['TotalBsmtSF']),fit = norm)
plt.figure()

_ = stats.probplot(np.log(df_train[df_train['TotalBsmtSF']!=0]['TotalBsmtSF']),plot = plt)
```



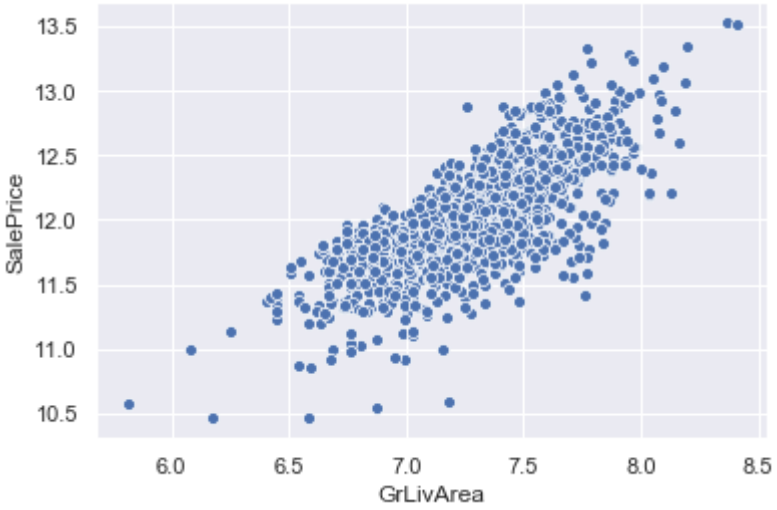


In [206]:

```
sns.scatterplot(np.log(df_train['GrLivArea']),np.log(df_train['SalePrice']))
```

Out[206]:

<matplotlib.axes._subplots.AxesSubplot at 0xb8546a0>



In [208]:

```
plt.scatter(np.log(df_train[df_train['TotalBsmtSF']>0]['TotalBsmtSF']), np.log(df_train[df_train['To
```