house_price_ana_vis_regression

December 11, 2017

In [1]:
```
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python docker image: https://github.com/kaggle/docker-pyth
# For example, here's several helpful packages to load in

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list the file

from subprocess import check_output
print(check_output(["ls", "../input"]).decode("utf8"))

# Any results you write to the current directory are saved as output.
```

```
sample_submission.csv
test.csv
train.csv
```

In [2]:
```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list the file

from subprocess import check_output
print(check_output(["ls", "../input"]).decode("utf8"))
```

```
sample_submission.csv
test.csv
train.csv
```

In [3]:
```
test_df = pd.read_csv('../input/test.csv')
test_df.head()
```

Out[3]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	\
0	1461	20	RH	80.0	11622	Pave	NaN	Reg	
1	1462	20	RL	81.0	14267	Pave	NaN	IR1	
2	1463	60	RL	74.0	13830	Pave	NaN	IR1	
3	1464	60	RL	78.0	9978	Pave	NaN	IR1	
4	1465	120	RL	43.0	5005	Pave	NaN	IR1	

	LandContour	Utilities	...	ScreenPorch	PoolArea	PoolQC	Fence	\
0	Lvl	AllPub	...	120	0	NaN	MnPrv	
1	Lvl	AllPub	...	0	0	NaN	NaN	
2	Lvl	AllPub	...	0	0	NaN	MnPrv	
3	Lvl	AllPub	...	0	0	NaN	NaN	
4	HLS	AllPub	...	144	0	NaN	NaN	

	MiscFeature	MiscVal	MoSold	YrSold	SaleType	SaleCondition
0	NaN	0	6	2010	WD	Normal
1	Gar2	12500	6	2010	WD	Normal
2	NaN	0	3	2010	WD	Normal
3	NaN	0	6	2010	WD	Normal
4	NaN	0	1	2010	WD	Normal

[5 rows x 80 columns]

In [4]:
```
from scipy import stats
import seaborn as sns
```

In [5]:
```
train_df = pd.read_csv('../input/train.csv')
train_df.head()
```

Out[5]:

	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	\
0	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
1	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
2	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
3	Lvl	AllPub	...	0	NaN	NaN	NaN	0	
4	Lvl	AllPub	...	0	NaN	NaN	NaN	0	

	MoSold	YrSold	SaleType	SaleCondition	SalePrice
0	2	2008	WD	Normal	208500
1	5	2007	WD	Normal	181500
2	9	2008	WD	Normal	223500
3	2	2006	WD	Abnorml	140000

```
4     12   2008        WD         Normal     250000

[5 rows x 81 columns]
```

In [6]:
```
train_df.count
```

Out[6]:
```
<bound method DataFrame.count of         Id  MSSubClass MSZoning  LotFrontage  LotArea S
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
5        6          50       RL         85.0    14115   Pave   NaN      IR1
6        7          20       RL         75.0    10084   Pave   NaN      Reg
7        8          60       RL          NaN    10382   Pave   NaN      IR1
8        9          50       RM         51.0     6120   Pave   NaN      Reg
9       10         190       RL         50.0     7420   Pave   NaN      Reg
10      11          20       RL         70.0    11200   Pave   NaN      Reg
11      12          60       RL         85.0    11924   Pave   NaN      IR1
12      13          20       RL          NaN    12968   Pave   NaN      IR2
13      14          20       RL         91.0    10652   Pave   NaN      IR1
14      15          20       RL          NaN    10920   Pave   NaN      IR1
15      16          45       RM         51.0     6120   Pave   NaN      Reg
16      17          20       RL          NaN    11241   Pave   NaN      IR1
17      18          90       RL         72.0    10791   Pave   NaN      Reg
18      19          20       RL         66.0    13695   Pave   NaN      Reg
19      20          20       RL         70.0     7560   Pave   NaN      Reg
20      21          60       RL        101.0    14215   Pave   NaN      IR1
21      22          45       RM         57.0     7449   Pave  Grvl      Reg
22      23          20       RL         75.0     9742   Pave   NaN      Reg
23      24         120       RM         44.0     4224   Pave   NaN      Reg
24      25          20       RL          NaN     8246   Pave   NaN      IR1
25      26          20       RL        110.0    14230   Pave   NaN      Reg
26      27          20       RL         60.0     7200   Pave   NaN      Reg
27      28          20       RL         98.0    11478   Pave   NaN      Reg
28      29          20       RL         47.0    16321   Pave   NaN      IR1
29      30          30       RM         60.0     6324   Pave   NaN      IR1
...    ...         ...      ...          ...      ...    ...   ...      ...
1430  1431          60       RL         60.0    21930   Pave   NaN      IR3
1431  1432         120       RL          NaN     4928   Pave   NaN      IR1
1432  1433          30       RL         60.0    10800   Pave  Grvl      Reg
1433  1434          60       RL         93.0    10261   Pave   NaN      IR1
1434  1435          20       RL         80.0    17400   Pave   NaN      Reg
1435  1436          20       RL         80.0     8400   Pave   NaN      Reg
1436  1437          20       RL         60.0     9000   Pave   NaN      Reg
1437  1438          20       RL         96.0    12444   Pave   NaN      Reg
1438  1439          20       RM         90.0     7407   Pave   NaN      Reg
1439  1440          60       RL         80.0    11584   Pave   NaN      Reg
```

1440	1441	70	RL	79.0	11526	Pave	NaN	IR1
1441	1442	120	RM	NaN	4426	Pave	NaN	Reg
1442	1443	60	FV	85.0	11003	Pave	NaN	Reg
1443	1444	30	RL	NaN	8854	Pave	NaN	Reg
1444	1445	20	RL	63.0	8500	Pave	NaN	Reg
1445	1446	85	RL	70.0	8400	Pave	NaN	Reg
1446	1447	20	RL	NaN	26142	Pave	NaN	IR1
1447	1448	60	RL	80.0	10000	Pave	NaN	Reg
1448	1449	50	RL	70.0	11767	Pave	NaN	Reg
1449	1450	180	RM	21.0	1533	Pave	NaN	Reg
1450	1451	90	RL	60.0	9000	Pave	NaN	Reg
1451	1452	20	RL	78.0	9262	Pave	NaN	Reg
1452	1453	180	RM	35.0	3675	Pave	NaN	Reg
1453	1454	20	RL	90.0	17217	Pave	NaN	Reg
1454	1455	20	FV	62.0	7500	Pave	Pave	Reg
1455	1456	60	RL	62.0	7917	Pave	NaN	Reg
1456	1457	20	RL	85.0	13175	Pave	NaN	Reg
1457	1458	70	RL	66.0	9042	Pave	NaN	Reg
1458	1459	20	RL	68.0	9717	Pave	NaN	Reg
1459	1460	20	RL	75.0	9937	Pave	NaN	Reg

	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeature	\
0	Lvl	AllPub	...	0	NaN	NaN	NaN	
1	Lvl	AllPub	...	0	NaN	NaN	NaN	
2	Lvl	AllPub	...	0	NaN	NaN	NaN	
3	Lvl	AllPub	...	0	NaN	NaN	NaN	
4	Lvl	AllPub	...	0	NaN	NaN	NaN	
5	Lvl	AllPub	...	0	NaN	MnPrv	Shed	
6	Lvl	AllPub	...	0	NaN	NaN	NaN	
7	Lvl	AllPub	...	0	NaN	NaN	Shed	
8	Lvl	AllPub	...	0	NaN	NaN	NaN	
9	Lvl	AllPub	...	0	NaN	NaN	NaN	
10	Lvl	AllPub	...	0	NaN	NaN	NaN	
11	Lvl	AllPub	...	0	NaN	NaN	NaN	
12	Lvl	AllPub	...	0	NaN	NaN	NaN	
13	Lvl	AllPub	...	0	NaN	NaN	NaN	
14	Lvl	AllPub	...	0	NaN	GdWo	NaN	
15	Lvl	AllPub	...	0	NaN	GdPrv	NaN	
16	Lvl	AllPub	...	0	NaN	NaN	Shed	
17	Lvl	AllPub	...	0	NaN	NaN	Shed	
18	Lvl	AllPub	...	0	NaN	NaN	NaN	
19	Lvl	AllPub	...	0	NaN	MnPrv	NaN	
20	Lvl	AllPub	...	0	NaN	NaN	NaN	
21	Bnk	AllPub	...	0	NaN	GdPrv	NaN	
22	Lvl	AllPub	...	0	NaN	NaN	NaN	
23	Lvl	AllPub	...	0	NaN	NaN	NaN	
24	Lvl	AllPub	...	0	NaN	MnPrv	NaN	
25	Lvl	AllPub	...	0	NaN	NaN	NaN	

26	Lvl	AllPub	...	0	NaN	NaN	NaN
27	Lvl	AllPub	...	0	NaN	NaN	NaN
28	Lvl	AllPub	...	0	NaN	NaN	NaN
29	Lvl	AllPub	...	0	NaN	NaN	NaN
...	...	...	...	...	...	...	...
1430	Lvl	AllPub	...	0	NaN	NaN	NaN
1431	Lvl	AllPub	...	0	NaN	NaN	NaN
1432	Lvl	AllPub	...	0	NaN	NaN	NaN
1433	Lvl	AllPub	...	0	NaN	NaN	NaN
1434	Low	AllPub	...	0	NaN	NaN	NaN
1435	Lvl	AllPub	...	0	NaN	GdPrv	NaN
1436	Lvl	AllPub	...	0	NaN	GdWo	NaN
1437	Lvl	AllPub	...	0	NaN	NaN	NaN
1438	Lvl	AllPub	...	0	NaN	MnPrv	NaN
1439	Lvl	AllPub	...	0	NaN	NaN	NaN
1440	Bnk	AllPub	...	0	NaN	NaN	NaN
1441	Lvl	AllPub	...	0	NaN	NaN	NaN
1442	Lvl	AllPub	...	0	NaN	NaN	NaN
1443	Lvl	AllPub	...	0	NaN	NaN	NaN
1444	Lvl	AllPub	...	0	NaN	NaN	NaN
1445	Lvl	AllPub	...	0	NaN	NaN	NaN
1446	Lvl	AllPub	...	0	NaN	NaN	NaN
1447	Lvl	AllPub	...	0	NaN	NaN	NaN
1448	Lvl	AllPub	...	0	NaN	GdWo	NaN
1449	Lvl	AllPub	...	0	NaN	NaN	NaN
1450	Lvl	AllPub	...	0	NaN	NaN	NaN
1451	Lvl	AllPub	...	0	NaN	NaN	NaN
1452	Lvl	AllPub	...	0	NaN	NaN	NaN
1453	Lvl	AllPub	...	0	NaN	NaN	NaN
1454	Lvl	AllPub	...	0	NaN	NaN	NaN
1455	Lvl	AllPub	...	0	NaN	NaN	NaN
1456	Lvl	AllPub	...	0	NaN	MnPrv	NaN
1457	Lvl	AllPub	...	0	NaN	GdPrv	Shed
1458	Lvl	AllPub	...	0	NaN	NaN	NaN
1459	Lvl	AllPub	...	0	NaN	NaN	NaN

	MiscVal	MoSold	YrSold	SaleType	SaleCondition	SalePrice
0	0	2	2008	WD	Normal	208500
1	0	5	2007	WD	Normal	181500
2	0	9	2008	WD	Normal	223500
3	0	2	2006	WD	Abnorml	140000
4	0	12	2008	WD	Normal	250000
5	700	10	2009	WD	Normal	143000
6	0	8	2007	WD	Normal	307000
7	350	11	2009	WD	Normal	200000
8	0	4	2008	WD	Abnorml	129900
9	0	1	2008	WD	Normal	118000
10	0	2	2008	WD	Normal	129500

11	0	7	2006	New	Partial	345000
12	0	9	2008	WD	Normal	144000
13	0	8	2007	New	Partial	279500
14	0	5	2008	WD	Normal	157000
15	0	7	2007	WD	Normal	132000
16	700	3	2010	WD	Normal	149000
17	500	10	2006	WD	Normal	90000
18	0	6	2008	WD	Normal	159000
19	0	5	2009	COD	Abnorml	139000
20	0	11	2006	New	Partial	325300
21	0	6	2007	WD	Normal	139400
22	0	9	2008	WD	Normal	230000
23	0	6	2007	WD	Normal	129900
24	0	5	2010	WD	Normal	154000
25	0	7	2009	WD	Normal	256300
26	0	5	2010	WD	Normal	134800
27	0	5	2010	WD	Normal	306000
28	0	12	2006	WD	Normal	207500
29	0	5	2008	WD	Normal	68500
...	...	...	...	...	...	...
1430	0	7	2006	WD	Normal	192140
1431	0	10	2009	WD	Normal	143750
1432	0	8	2007	WD	Normal	64500
1433	0	5	2008	WD	Normal	186500
1434	0	5	2006	WD	Normal	160000
1435	0	7	2008	COD	Abnorml	174000
1436	0	5	2007	WD	Normal	120500
1437	0	11	2008	New	Partial	394617
1438	0	4	2010	WD	Normal	149700
1439	0	11	2007	WD	Normal	197000
1440	0	9	2008	WD	Normal	191000
1441	0	5	2008	WD	Normal	149300
1442	0	4	2009	WD	Normal	310000
1443	0	5	2009	WD	Normal	121000
1444	0	11	2007	WD	Normal	179600
1445	0	5	2007	WD	Normal	129000
1446	0	4	2010	WD	Normal	157900
1447	0	12	2007	WD	Normal	240000
1448	0	5	2007	WD	Normal	112000
1449	0	8	2006	WD	Abnorml	92000
1450	0	9	2009	WD	Normal	136000
1451	0	5	2009	New	Partial	287090
1452	0	5	2006	WD	Normal	145000
1453	0	7	2006	WD	Abnorml	84500
1454	0	10	2009	WD	Normal	185000
1455	0	8	2007	WD	Normal	175000
1456	0	2	2010	WD	Normal	210000
1457	2500	5	2010	WD	Normal	266500

```
1458       0      4   2010        WD         Normal     142125
1459       0      6   2008        WD         Normal     147500

[1460 rows x 81 columns]>
```

```
In [7]: train_df.shape[0]
```

```
Out[7]: 1460
```

```
In [8]: print(train_df.shape[1])
        train_df.columns
```

```
81
```

```
Out[8]: Index(['Id', 'MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea', 'Street',
               'Alley', 'LotShape', 'LandContour', 'Utilities', 'LotConfig',
               'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'BldgType',
               'HouseStyle', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd',
               'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType',
               'MasVnrArea', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual',
               'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1',
               'BsmtFinType2', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating',
               'HeatingQC', 'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF',
               'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath',
               'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual',
               'TotRmsAbvGrd', 'Functional', 'Fireplaces', 'FireplaceQu', 'GarageType',
               'GarageYrBlt', 'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQual',
               'GarageCond', 'PavedDrive', 'WoodDeckSF', 'OpenPorchSF',
               'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'PoolQC',
               'Fence', 'MiscFeature', 'MiscVal', 'MoSold', 'YrSold', 'SaleType',
               'SaleCondition', 'SalePrice'],
              dtype='object')
```

```
In [9]: test_df.isnull().sum()
```

```
Out[9]: Id                0
        MSSubClass        0
        MSZoning          4
        LotFrontage     227
        LotArea           0
        Street            0
        Alley          1352
        LotShape          0
        LandContour       0
        Utilities         2
        LotConfig         0
        LandSlope         0
        Neighborhood      0
```

```
Condition1          0
Condition2          0
BldgType            0
HouseStyle          0
OverallQual         0
OverallCond         0
YearBuilt           0
YearRemodAdd        0
RoofStyle           0
RoofMatl            0
Exterior1st         1
Exterior2nd         1
MasVnrType         16
MasVnrArea         15
ExterQual           0
ExterCond           0
Foundation          0
                  ...
HalfBath            0
BedroomAbvGr        0
KitchenAbvGr        0
KitchenQual         1
TotRmsAbvGrd        0
Functional          2
Fireplaces          0
FireplaceQu       730
GarageType         76
GarageYrBlt        78
GarageFinish       78
GarageCars          1
GarageArea          1
GarageQual         78
GarageCond         78
PavedDrive          0
WoodDeckSF          0
OpenPorchSF         0
EnclosedPorch       0
3SsnPorch           0
ScreenPorch         0
PoolArea            0
PoolQC           1456
Fence            1169
MiscFeature      1408
MiscVal             0
MoSold              0
YrSold              0
SaleType            1
SaleCondition       0
```

```
Length: 80, dtype: int64
```

In [10]:
```
null_val = pd.concat([train_df.isnull().sum(),train_df.isnull().sum()/train_df.shape[0]
                      test_df.isnull().sum(),train_df.isnull().sum()/test_df.shape[0]],
                     axis =1 ,keys = ['train_null_count','%_null','test_null_count','%_n
null_val[null_val.sum(axis=1)>0]
```

Out[10]:

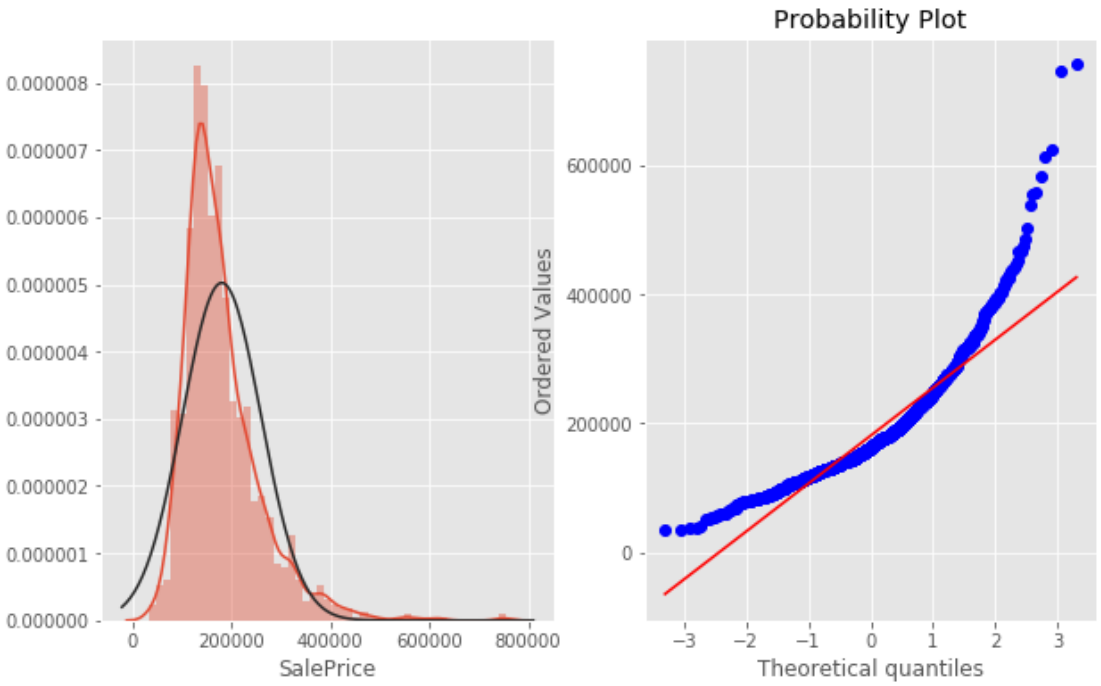
	train_null_count	%_null	test_null_count	%_null
Alley	1369	0.937671	1352.0	0.938314
BsmtCond	37	0.025342	45.0	0.025360
BsmtExposure	38	0.026027	44.0	0.026045
BsmtFinSF1	0	0.000000	1.0	0.000000
BsmtFinSF2	0	0.000000	1.0	0.000000
BsmtFinType1	37	0.025342	42.0	0.025360
BsmtFinType2	38	0.026027	42.0	0.026045
BsmtFullBath	0	0.000000	2.0	0.000000
BsmtHalfBath	0	0.000000	2.0	0.000000
BsmtQual	37	0.025342	44.0	0.025360
BsmtUnfSF	0	0.000000	1.0	0.000000
Electrical	1	0.000685	0.0	0.000685
Exterior1st	0	0.000000	1.0	0.000000
Exterior2nd	0	0.000000	1.0	0.000000
Fence	1179	0.807534	1169.0	0.808088
FireplaceQu	690	0.472603	730.0	0.472927
Functional	0	0.000000	2.0	0.000000
GarageArea	0	0.000000	1.0	0.000000
GarageCars	0	0.000000	1.0	0.000000
GarageCond	81	0.055479	78.0	0.055517
GarageFinish	81	0.055479	78.0	0.055517
GarageQual	81	0.055479	78.0	0.055517
GarageType	81	0.055479	76.0	0.055517
GarageYrBlt	81	0.055479	78.0	0.055517
KitchenQual	0	0.000000	1.0	0.000000
LotFrontage	259	0.177397	227.0	0.177519
MSZoning	0	0.000000	4.0	0.000000
MasVnrArea	8	0.005479	15.0	0.005483
MasVnrType	8	0.005479	16.0	0.005483
MiscFeature	1406	0.963014	1408.0	0.963674
PoolQC	1453	0.995205	1456.0	0.995888
SaleType	0	0.000000	1.0	0.000000
TotalBsmtSF	0	0.000000	1.0	0.000000
Utilities	0	0.000000	2.0	0.000000

In [11]:
```
train_df['SalePrice'].describe()
```

Out[11]:
```
count      1460.000000
mean     180921.195890
std       79442.502883
min       34900.000000
```

```
25%       129975.000000
50%       163000.000000
75%       214000.000000
max       755000.000000
Name: SalePrice, dtype: float64
```

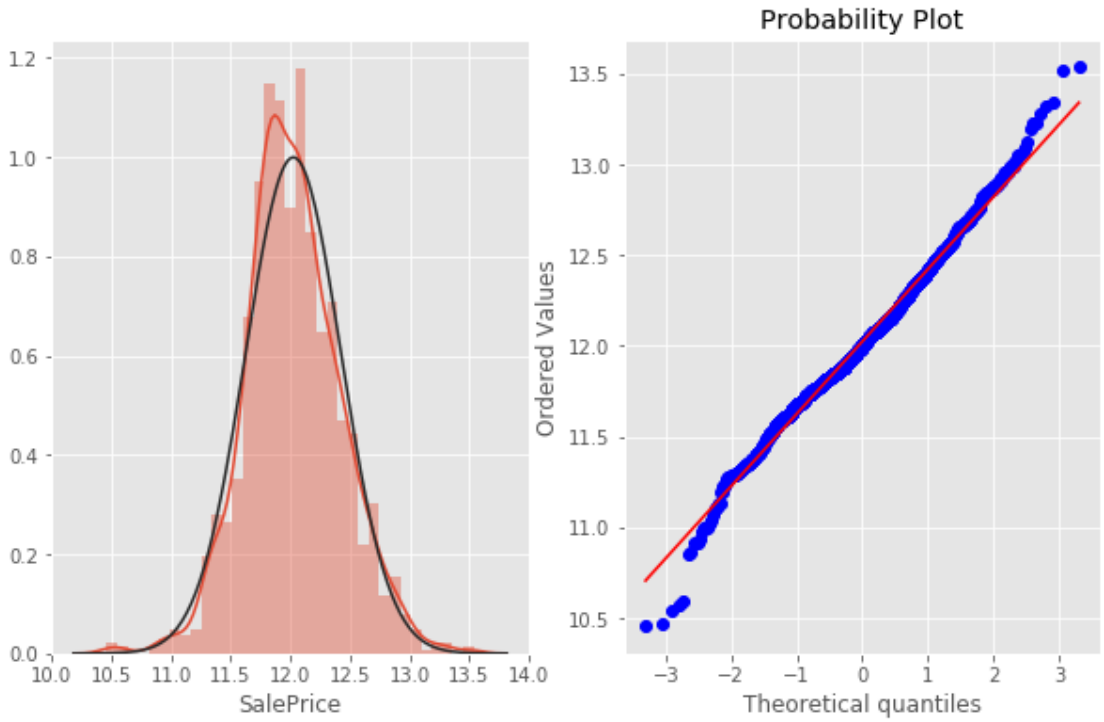
```
In [12]: import matplotlib.pyplot as plt
         plt.style.use(style='ggplot')
         plt.rcParams['figure.figsize'] = (10, 6)
         plt.figure()
         plt.subplot(1, 2, 1)
         sns.distplot(train_df['SalePrice'], fit=stats.norm)
         plt.subplot(1, 2, 2)
         stats.probplot(train_df['SalePrice'], plot=plt)
         plt.show()
         print("Skewness: %f" % train_df['SalePrice'].skew())
         print("Kurtosis: %f" % train_df['SalePrice'].kurt())
```




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

```
In [13]: plt.figure()
         plt.subplot(1, 2, 1)
         sns.distplot(np.log(train_df['SalePrice']+1), fit=stats.norm)
```

```
plt.subplot(1, 2, 2)
stats.probplot(np.log(train_df['SalePrice']+1), plot=plt)
plt.show()
print("Skewness: %f" % np.log(train_df['SalePrice']+1).skew())
print("Kurtosis: %f" % np.log(train_df['SalePrice']+1).kurt())
```




```
Skewness: 0.121347
Kurtosis: 0.809519
```

In [14]:
```
corrmat = train_df.corr()
corrmat
```

Out[14]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	\
Id	1.000000	0.011156	-0.010601	-0.033226	-0.028365	
MSSubClass	0.011156	1.000000	-0.386347	-0.139781	0.032628	
LotFrontage	-0.010601	-0.386347	1.000000	0.426095	0.251646	
LotArea	-0.033226	-0.139781	0.426095	1.000000	0.105806	
OverallQual	-0.028365	0.032628	0.251646	0.105806	1.000000	
OverallCond	0.012609	-0.059316	-0.059213	-0.005636	-0.091932	
YearBuilt	-0.012713	0.027850	0.123349	0.014228	0.572323	
YearRemodAdd	-0.021998	0.040581	0.088866	0.013788	0.550684	
MasVnrArea	-0.050298	0.022936	0.193458	0.104160	0.411876	
BsmtFinSF1	-0.005024	-0.069836	0.233633	0.214103	0.239666	

BsmtFinSF2	-0.005968	-0.065649	0.049900	0.111170	-0.059119
BsmtUnfSF	-0.007940	-0.140759	0.132644	-0.002618	0.308159
TotalBsmtSF	-0.015415	-0.238518	0.392075	0.260833	0.537808
1stFlrSF	0.010496	-0.251758	0.457181	0.299475	0.476224
2ndFlrSF	0.005590	0.307886	0.080177	0.050986	0.295493
LowQualFinSF	-0.044230	0.046474	0.038469	0.004779	-0.030429
GrLivArea	0.008273	0.074853	0.402797	0.263116	0.593007
BsmtFullBath	0.002289	0.003491	0.100949	0.158155	0.111098
BsmtHalfBath	-0.020155	-0.002333	-0.007234	0.048046	-0.040150
FullBath	0.005587	0.131608	0.198769	0.126031	0.550600
HalfBath	0.006784	0.177354	0.053532	0.014259	0.273458
BedroomAbvGr	0.037719	-0.023438	0.263170	0.119690	0.101676
KitchenAbvGr	0.002951	0.281721	-0.006069	-0.017784	-0.183882
TotRmsAbvGrd	0.027239	0.040380	0.352096	0.190015	0.427452
Fireplaces	-0.019772	-0.045569	0.266639	0.271364	0.396765
GarageYrBlt	0.000072	0.085072	0.070250	-0.024947	0.547766
GarageCars	0.016570	-0.040110	0.285691	0.154871	0.600671
GarageArea	0.017634	-0.098672	0.344997	0.180403	0.562022
WoodDeckSF	-0.029643	-0.012579	0.088521	0.171698	0.238923
OpenPorchSF	-0.000477	-0.006100	0.151972	0.084774	0.308819
EnclosedPorch	0.002889	-0.012037	0.010700	-0.018340	-0.113937
3SsnPorch	-0.046635	-0.043825	0.070029	0.020423	0.030371
ScreenPorch	0.001330	-0.026030	0.041383	0.043160	0.064886
PoolArea	0.057044	0.008283	0.206167	0.077672	0.065166
MiscVal	-0.006242	-0.007683	0.003368	0.038068	-0.031406
MoSold	0.021172	-0.013585	0.011200	0.001205	0.070815
YrSold	0.000712	-0.021407	0.007450	-0.014261	-0.027347
SalePrice	-0.021917	-0.084284	0.351799	0.263843	0.790982

	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	\
Id	0.012609	-0.012713	-0.021998	-0.050298	-0.005024	
MSSubClass	-0.059316	0.027850	0.040581	0.022936	-0.069836	
LotFrontage	-0.059213	0.123349	0.088866	0.193458	0.233633	
LotArea	-0.005636	0.014228	0.013788	0.104160	0.214103	
OverallQual	-0.091932	0.572323	0.550684	0.411876	0.239666	
OverallCond	1.000000	-0.375983	0.073741	-0.128101	-0.046231	
YearBuilt	-0.375983	1.000000	0.592855	0.315707	0.249503	
YearRemodAdd	0.073741	0.592855	1.000000	0.179618	0.128451	
MasVnrArea	-0.128101	0.315707	0.179618	1.000000	0.264736	
BsmtFinSF1	-0.046231	0.249503	0.128451	0.264736	1.000000	
BsmtFinSF2	0.040229	-0.049107	-0.067759	-0.072319	-0.050117	
BsmtUnfSF	-0.136841	0.149040	0.181133	0.114442	-0.495251	
TotalBsmtSF	-0.171098	0.391452	0.291066	0.363936	0.522396	
1stFlrSF	-0.144203	0.281986	0.240379	0.344501	0.445863	
2ndFlrSF	0.028942	0.010308	0.140024	0.174561	-0.137079	
LowQualFinSF	0.025494	-0.183784	-0.062419	-0.069071	-0.064503	
GrLivArea	-0.079686	0.199010	0.287389	0.390857	0.208171	
BsmtFullBath	-0.054942	0.187599	0.119470	0.085310	0.649212	

BsmtHalfBath	0.117821	-0.038162	-0.012337	0.026673	0.067418
FullBath	-0.194149	0.468271	0.439046	0.276833	0.058543
HalfBath	-0.060769	0.242656	0.183331	0.201444	0.004262
BedroomAbvGr	0.012980	-0.070651	-0.040581	0.102821	-0.107355
KitchenAbvGr	-0.087001	-0.174800	-0.149598	-0.037610	-0.081007
TotRmsAbvGrd	-0.057583	0.095589	0.191740	0.280682	0.044316
Fireplaces	-0.023820	0.147716	0.112581	0.249070	0.260011
GarageYrBlt	-0.324297	0.825667	0.642277	0.252691	0.153484
GarageCars	-0.185758	0.537850	0.420622	0.364204	0.224054
GarageArea	-0.151521	0.478954	0.371600	0.373066	0.296970
WoodDeckSF	-0.003334	0.224880	0.205726	0.159718	0.204306
OpenPorchSF	-0.032589	0.188686	0.226298	0.125703	0.111761
EnclosedPorch	0.070356	-0.387268	-0.193919	-0.110204	-0.102303
3SsnPorch	0.025504	0.031355	0.045286	0.018796	0.026451
ScreenPorch	0.054811	-0.050364	-0.038740	0.061466	0.062021
PoolArea	-0.001985	0.004950	0.005829	0.011723	0.140491
MiscVal	0.068777	-0.034383	-0.010286	-0.029815	0.003571
MoSold	-0.003511	0.012398	0.021490	-0.005965	-0.015727
YrSold	0.043950	-0.013618	0.035743	-0.008201	0.014359
SalePrice	-0.077856	0.522897	0.507101	0.477493	0.386420

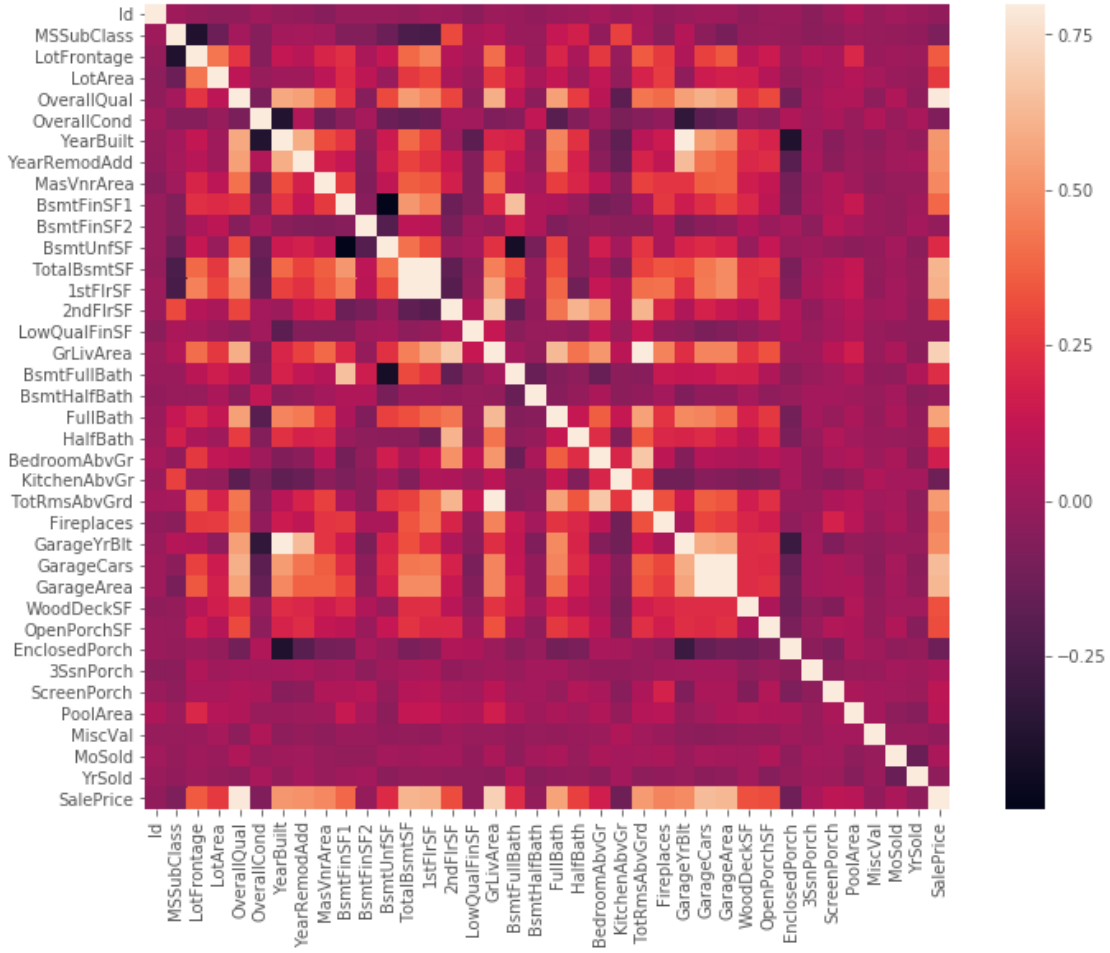
	...	WoodDeckSF	OpenPorchSF	EnclosedPorch	3SsnPorch	\
Id	...	-0.029643	-0.000477	0.002889	-0.046635	
MSSubClass	...	-0.012579	-0.006100	-0.012037	-0.043825	
LotFrontage	...	0.088521	0.151972	0.010700	0.070029	
LotArea	...	0.171698	0.084774	-0.018340	0.020423	
OverallQual	...	0.238923	0.308819	-0.113937	0.030371	
OverallCond	...	-0.003334	-0.032589	0.070356	0.025504	
YearBuilt	...	0.224880	0.188686	-0.387268	0.031355	
YearRemodAdd	...	0.205726	0.226298	-0.193919	0.045286	
MasVnrArea	...	0.159718	0.125703	-0.110204	0.018796	
BsmtFinSF1	...	0.204306	0.111761	-0.102303	0.026451	
BsmtFinSF2	...	0.067898	0.003093	0.036543	-0.029993	
BsmtUnfSF	...	-0.005316	0.129005	-0.002538	0.020764	
TotalBsmtSF	...	0.232019	0.247264	-0.095478	0.037384	
1stFlrSF	...	0.235459	0.211671	-0.065292	0.056104	
2ndFlrSF	...	0.092165	0.208026	0.061989	-0.024358	
LowQualFinSF	...	-0.025444	0.018251	0.061081	-0.004296	
GrLivArea	...	0.247433	0.330224	0.009113	0.020643	
BsmtFullBath	...	0.175315	0.067341	-0.049911	-0.000106	
BsmtHalfBath	...	0.040161	-0.025324	-0.008555	0.035114	
FullBath	...	0.187703	0.259977	-0.115093	0.035353	
HalfBath	...	0.108080	0.199740	-0.095317	-0.004972	
BedroomAbvGr	...	0.046854	0.093810	0.041570	-0.024478	
KitchenAbvGr	...	-0.090130	-0.070091	0.037312	-0.024600	
TotRmsAbvGrd	...	0.165984	0.234192	0.004151	-0.006683	
Fireplaces	...	0.200019	0.169405	-0.024822	0.011257	
GarageYrBlt	...	0.224577	0.228425	-0.297003	0.023544	

GarageCars	...	0.226342	0.213569	-0.151434	0.035765
GarageArea	...	0.224666	0.241435	-0.121777	0.035087
WoodDeckSF	...	1.000000	0.058661	-0.125989	-0.032771
OpenPorchSF	...	0.058661	1.000000	-0.093079	-0.005842
EnclosedPorch	...	-0.125989	-0.093079	1.000000	-0.037305
3SsnPorch	...	-0.032771	-0.005842	-0.037305	1.000000
ScreenPorch	...	-0.074181	0.074304	-0.082864	-0.031436
PoolArea	...	0.073378	0.060762	0.054203	-0.007992
MiscVal	...	-0.009551	-0.018584	0.018361	0.000354
MoSold	...	0.021011	0.071255	-0.028887	0.029474
YrSold	...	0.022270	-0.057619	-0.009916	0.018645
SalePrice	...	0.324413	0.315856	-0.128578	0.044584

	ScreenPorch	PoolArea	MiscVal	MoSold	YrSold	SalePrice
Id	0.001330	0.057044	-0.006242	0.021172	0.000712	-0.021917
MSSubClass	-0.026030	0.008283	-0.007683	-0.013585	-0.021407	-0.084284
LotFrontage	0.041383	0.206167	0.003368	0.011200	0.007450	0.351799
LotArea	0.043160	0.077672	0.038068	0.001205	-0.014261	0.263843
OverallQual	0.064886	0.065166	-0.031406	0.070815	-0.027347	0.790982
OverallCond	0.054811	-0.001985	0.068777	-0.003511	0.043950	-0.077856
YearBuilt	-0.050364	0.004950	-0.034383	0.012398	-0.013618	0.522897
YearRemodAdd	-0.038740	0.005829	-0.010286	0.021490	0.035743	0.507101
MasVnrArea	0.061466	0.011723	-0.029815	-0.005965	-0.008201	0.477493
BsmtFinSF1	0.062021	0.140491	0.003571	-0.015727	0.014359	0.386420
BsmtFinSF2	0.088871	0.041709	0.004940	-0.015211	0.031706	-0.011378
BsmtUnfSF	-0.012579	-0.035092	-0.023837	0.034888	-0.041258	0.214479
TotalBsmtSF	0.084489	0.126053	-0.018479	0.013196	-0.014969	0.613581
1stFlrSF	0.088758	0.131525	-0.021096	0.031372	-0.013604	0.605852
2ndFlrSF	0.040606	0.081487	0.016197	0.035164	-0.028700	0.319334
LowQualFinSF	0.026799	0.062157	-0.003793	-0.022174	-0.028921	-0.025606
GrLivArea	0.101510	0.170205	-0.002416	0.050240	-0.036526	0.708624
BsmtFullBath	0.023148	0.067616	-0.023047	-0.025361	0.067049	0.227122
BsmtHalfBath	0.032121	0.020025	-0.007367	0.032873	-0.046524	-0.016844
FullBath	-0.008106	0.049604	-0.014290	0.055872	-0.019669	0.560664
HalfBath	0.072426	0.022381	0.001290	-0.009050	-0.010269	0.284108
BedroomAbvGr	0.044300	0.070703	0.007767	0.046544	-0.036014	0.168213
KitchenAbvGr	-0.051613	-0.014525	0.062341	0.026589	0.031687	-0.135907
TotRmsAbvGrd	0.059383	0.083757	0.024763	0.036907	-0.034516	0.533723
Fireplaces	0.184530	0.095074	0.001409	0.046357	-0.024096	0.466929
GarageYrBlt	-0.075418	-0.014501	-0.032417	0.005337	-0.001014	0.486362
GarageCars	0.050494	0.020934	-0.043080	0.040522	-0.039117	0.640409
GarageArea	0.051412	0.061047	-0.027400	0.027974	-0.027378	0.623431
WoodDeckSF	-0.074181	0.073378	-0.009551	0.021011	0.022270	0.324413
OpenPorchSF	0.074304	0.060762	-0.018584	0.071255	-0.057619	0.315856
EnclosedPorch	-0.082864	0.054203	0.018361	-0.028887	-0.009916	-0.128578
3SsnPorch	-0.031436	-0.007992	0.000354	0.029474	0.018645	0.044584
ScreenPorch	1.000000	0.051307	0.031946	0.023217	0.010694	0.111447
PoolArea	0.051307	1.000000	0.029669	-0.033737	-0.059689	0.092404

```
MiscVal          0.031946  0.029669  1.000000 -0.006495  0.004906 -0.021190
MoSold           0.023217 -0.033737 -0.006495  1.000000 -0.145721  0.046432
YrSold           0.010694 -0.059689  0.004906 -0.145721  1.000000 -0.028923
SalePrice        0.111447  0.092404 -0.021190  0.046432 -0.028923  1.000000

[38 rows x 38 columns]
```

In [15]:
```
f, ax = plt.subplots(figsize=(12, 9))
sns.heatmap(corrmat, vmax=.8, square=True)
plt.yticks(rotation=0)
plt.xticks(rotation=90)
plt.show()
```

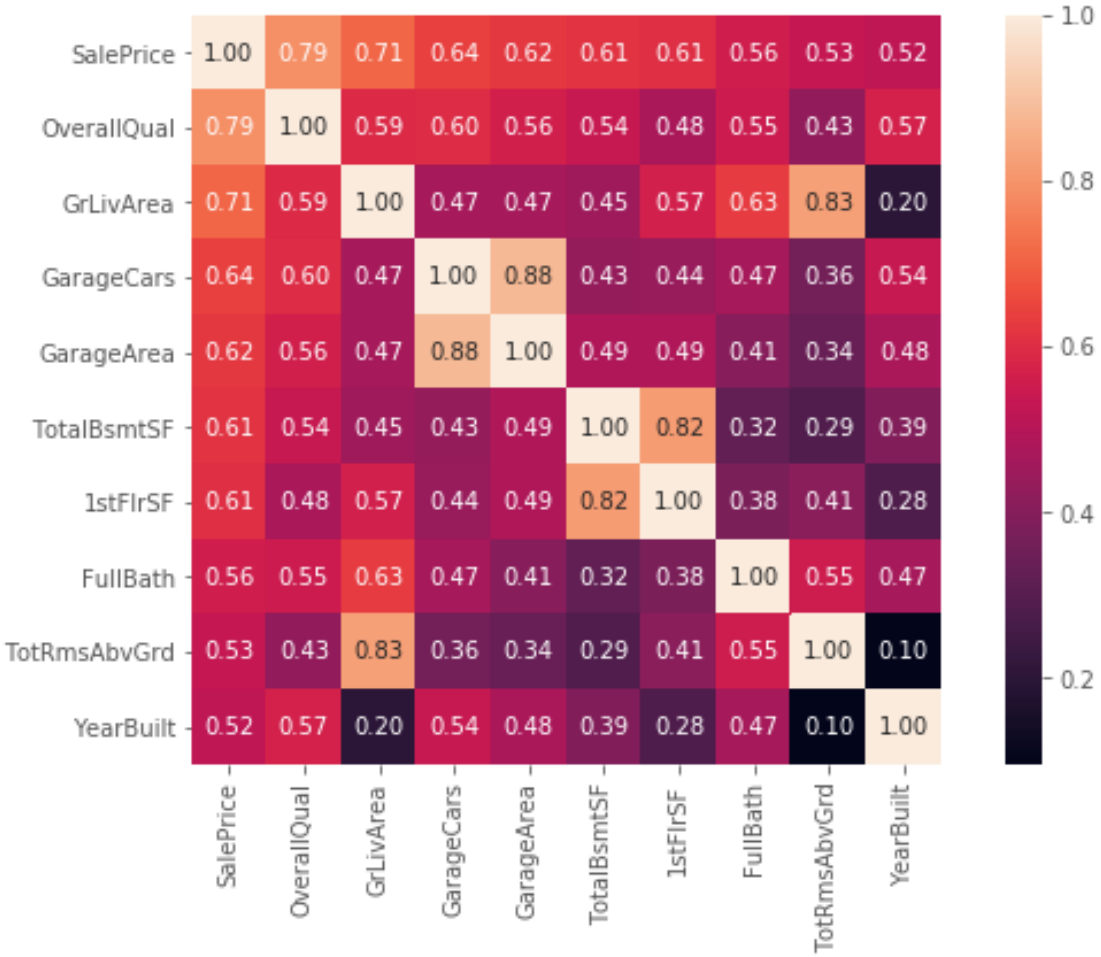


In [16]:
```
cols = corrmat.nlargest(10, 'SalePrice')['SalePrice'].index
```

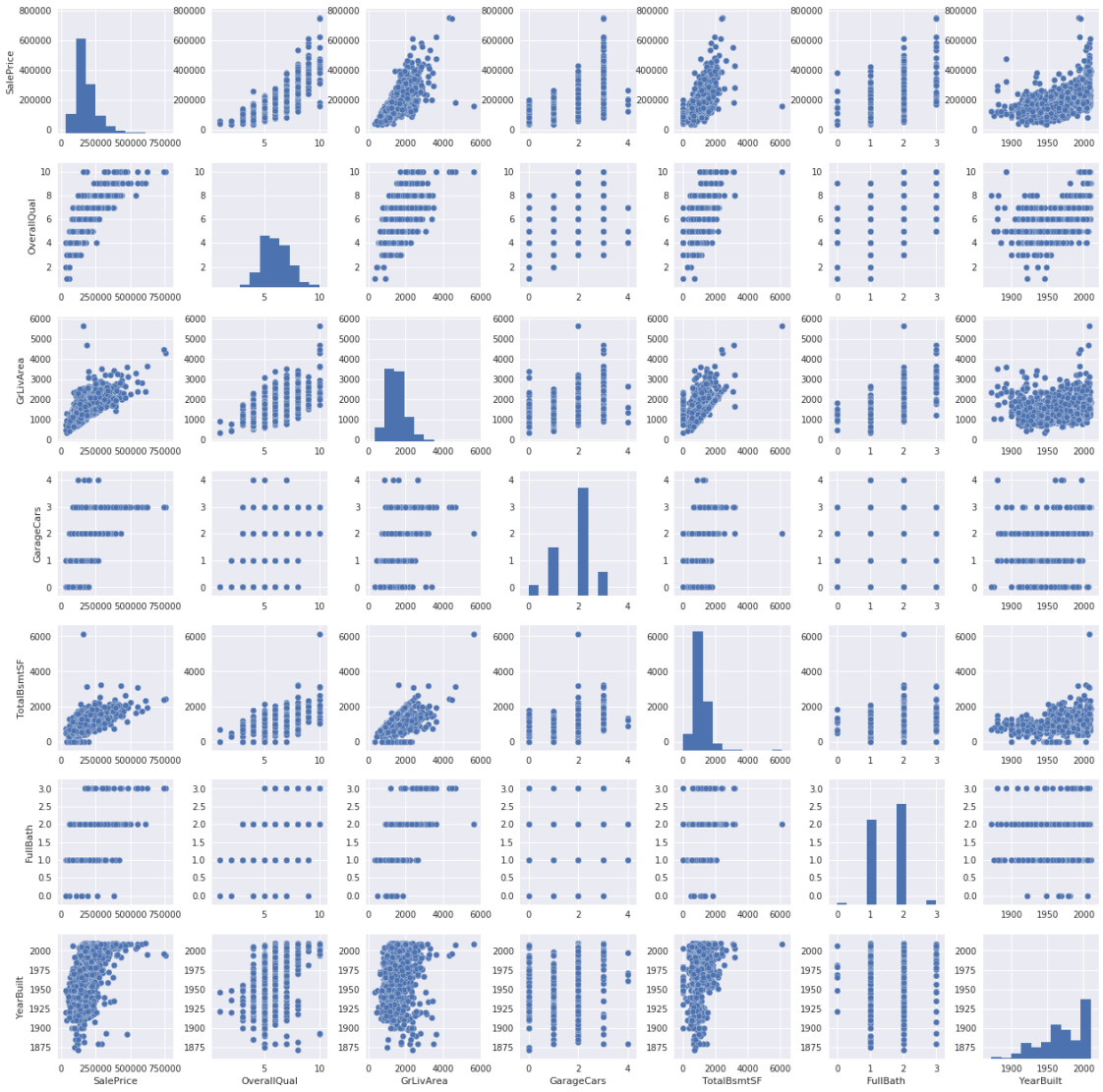
In [17]:
```
cols
```

```
Out[17]: Index(['SalePrice', 'OverallQual', 'GrLivArea', 'GarageCars', 'GarageArea',
                'TotalBsmtSF', '1stFlrSF', 'FullBath', 'TotRmsAbvGrd', 'YearBuilt'],
               dtype='object')
```

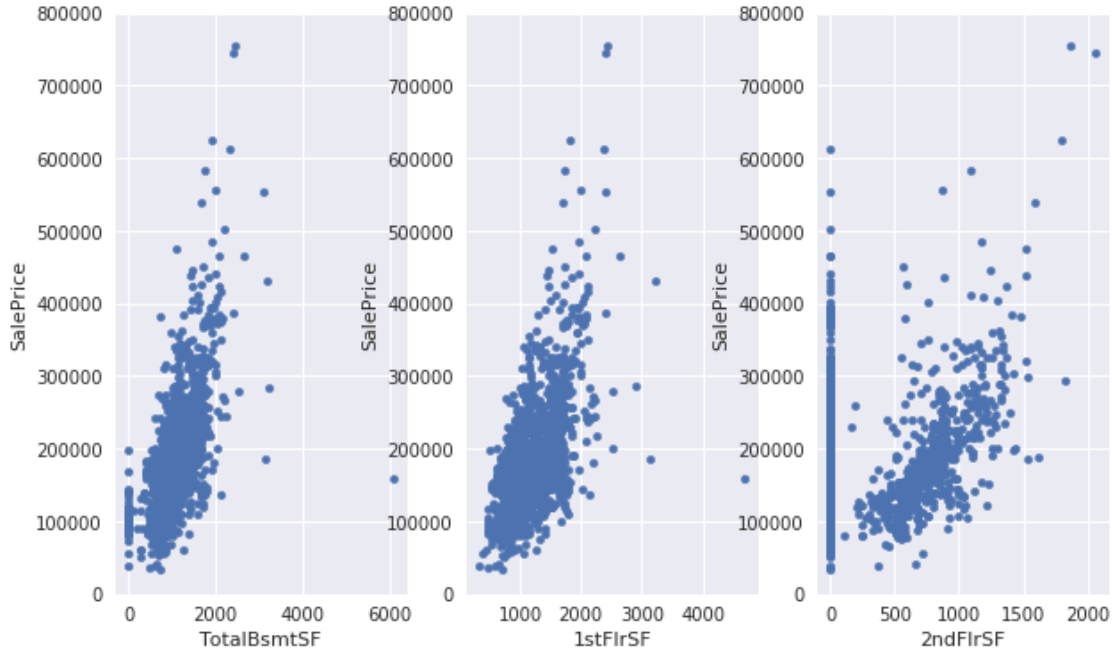
```
In [18]: cm = np.corrcoef(train_df[cols].values.T)
         hm = sns.heatmap(cm, cbar=True, annot=True, square=True, fmt='.2f', annot_kws={'size':
                          xticklabels=cols.values)
         plt.yticks(rotation=0)
         plt.xticks(rotation=90)
         plt.show()
```



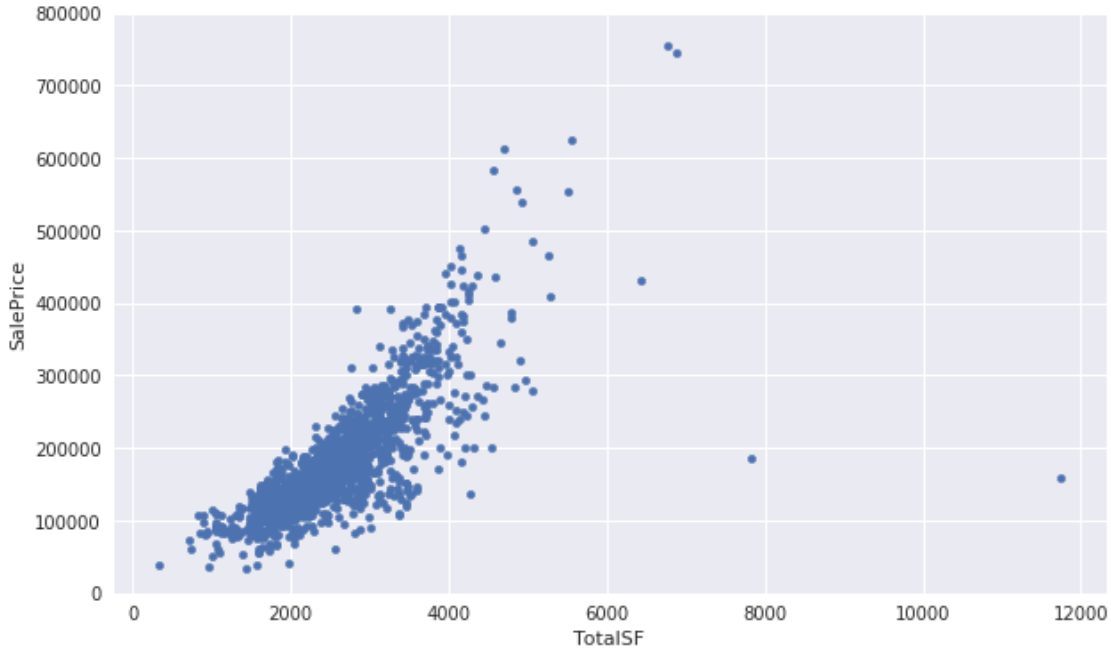

```
In [19]: sns.set()
         cols = ['SalePrice', 'OverallQual', 'GrLivArea', 'GarageCars', 'TotalBsmtSF', 'FullBath
         sns.pairplot(train_df[cols], size=2.5)
         plt.show()
```



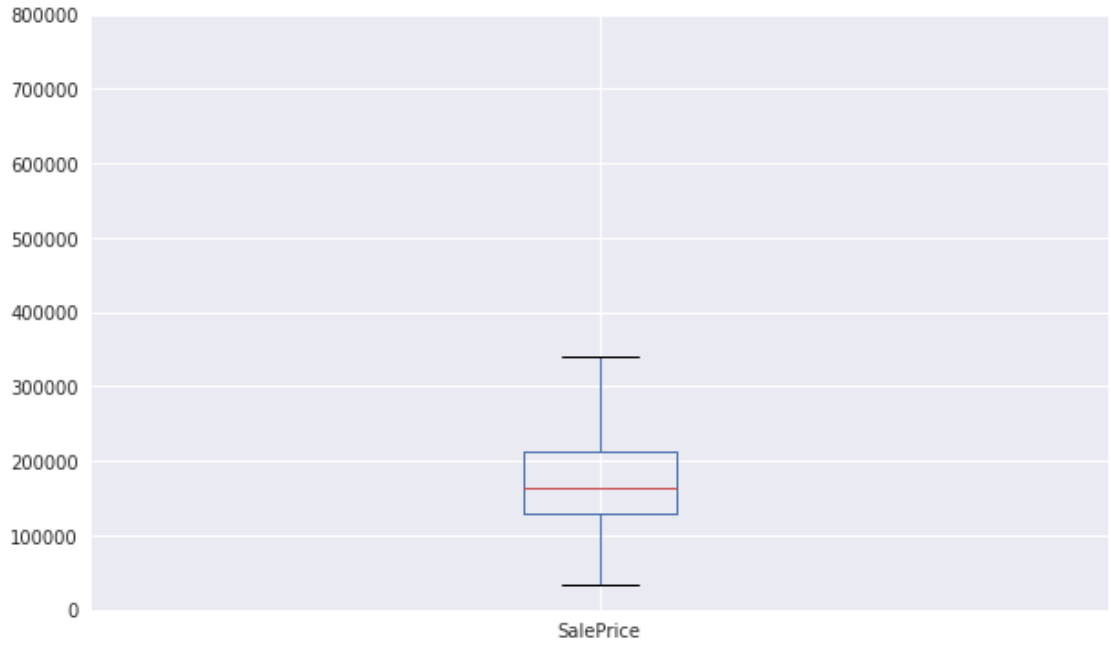

```
In [20]: f, (ax1, ax2, ax3) = plt.subplots(1, 3)
         data_total = pd.concat([train_df['SalePrice'], train_df['TotalBsmtSF']], axis=1)
         data_total.plot.scatter(x='TotalBsmtSF', y='SalePrice', ylim=(0, 800000), ax=ax1)
         data1 = pd.concat([train_df['SalePrice'], train_df['1stFlrSF']], axis=1)
         data1.plot.scatter(x='1stFlrSF', y='SalePrice', ylim=(0, 800000), ax=ax2)
         data2 = pd.concat([train_df['SalePrice'], train_df['2ndFlrSF']], axis=1)
         data2.plot.scatter(x='2ndFlrSF', y='SalePrice', ylim=(0, 800000), ax=ax3)
         plt.show()
```



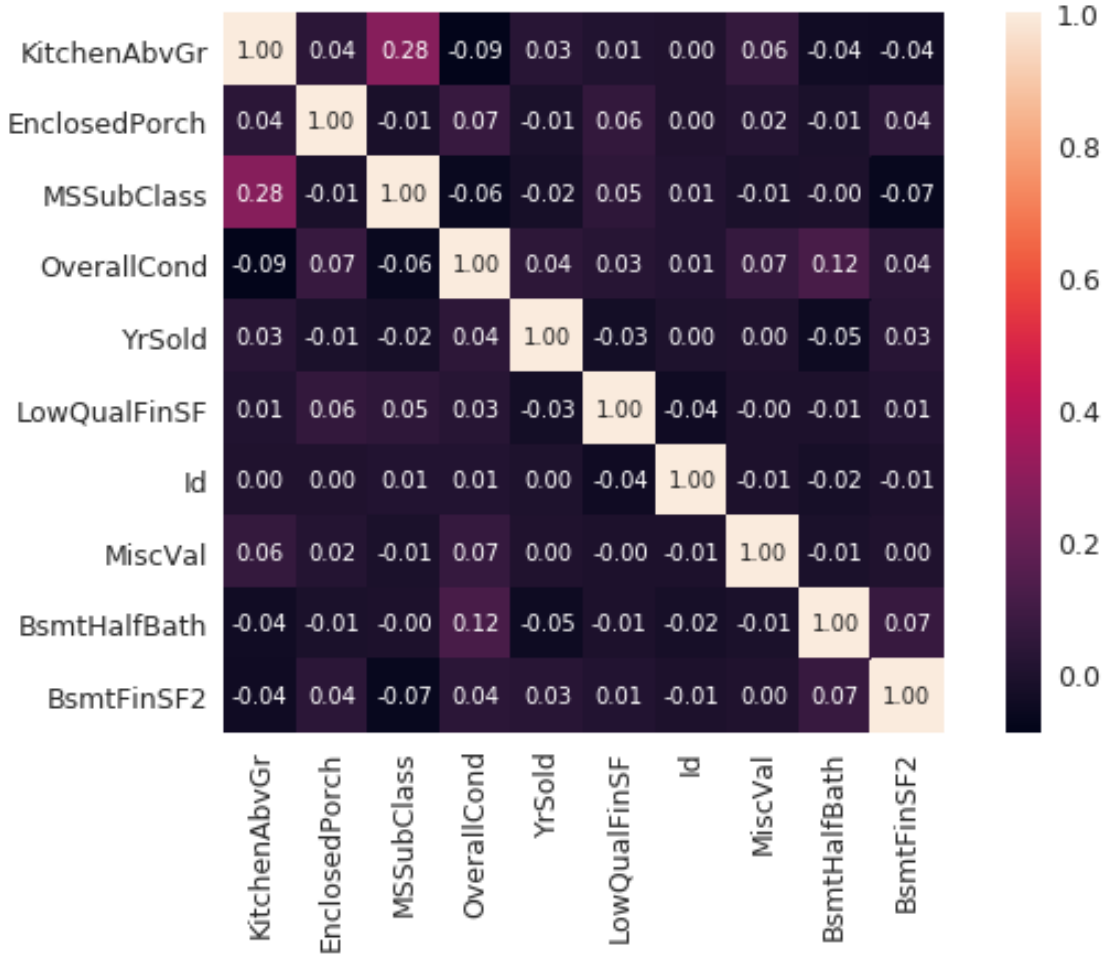

```
In [21]: train_df['TotalBsmtSF'] = train_df['TotalBsmtSF'].fillna(0)
         train_df['1stFlrSF'] = train_df['1stFlrSF'].fillna(0)
         train_df['2ndFlrSF'] = train_df['2ndFlrSF'].fillna(0)
         train_df['TotalSF'] = train_df['TotalBsmtSF'] + train_df['1stFlrSF'] + train_df['2ndFlr

In [22]: data2 = pd.concat([train_df['SalePrice'], train_df['TotalSF']], axis=1)
         data2.plot.scatter(x='TotalSF', y='SalePrice', ylim=(0, 800000))
         plt.show()
```



In [23]:
```
data2 = pd.concat([train_df['SalePrice'], train_df['TotalSF']], axis=1)
data2.plot.box(x='TotalSF', y='SalePrice', ylim=(0, 800000))
plt.show()
```



In [24]:
```
corrmat = train_df.corr()
cols = corrmat.nsmallest(10,'SalePrice')['SalePrice'].index
cm = np.corrcoef(train_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':
                 xticklabels=cols.values)
plt.yticks(rotation=0)
plt.xticks(rotation=90)
plt.show()
```



In [25]:
```
corrmat = train_df.corr()
cols = corrmat.nsmallest(20,'SalePrice')['SalePrice'].index
cols
```

Out[25]:
```
Index(['KitchenAbvGr', 'EnclosedPorch', 'MSSubClass', 'OverallCond', 'YrSold',
       'LowQualFinSF', 'Id', 'MiscVal', 'BsmtHalfBath', 'BsmtFinSF2',
       '3SsnPorch', 'MoSold', 'PoolArea', 'ScreenPorch', 'BedroomAbvGr',
```

```
       'BsmtUnfSF', 'BsmtFullBath', 'LotArea', 'HalfBath', 'OpenPorchSF'],
      dtype='object')
```

In [26]:
```
corrmat = train_df.corr()
corrmat.nsmallest(10,'SalePrice')['SalePrice']
```

Out[26]:
```
KitchenAbvGr    -0.135907
EnclosedPorch   -0.128578
MSSubClass      -0.084284
OverallCond     -0.077856
YrSold          -0.028923
LowQualFinSF    -0.025606
Id              -0.021917
MiscVal         -0.021190
BsmtHalfBath    -0.016844
BsmtFinSF2      -0.011378
Name: SalePrice, dtype: float64
```

In [27]:
```
corrmat = train_df.corr().abs()
corrmat.nsmallest(10,'SalePrice')
```

Out[27]:

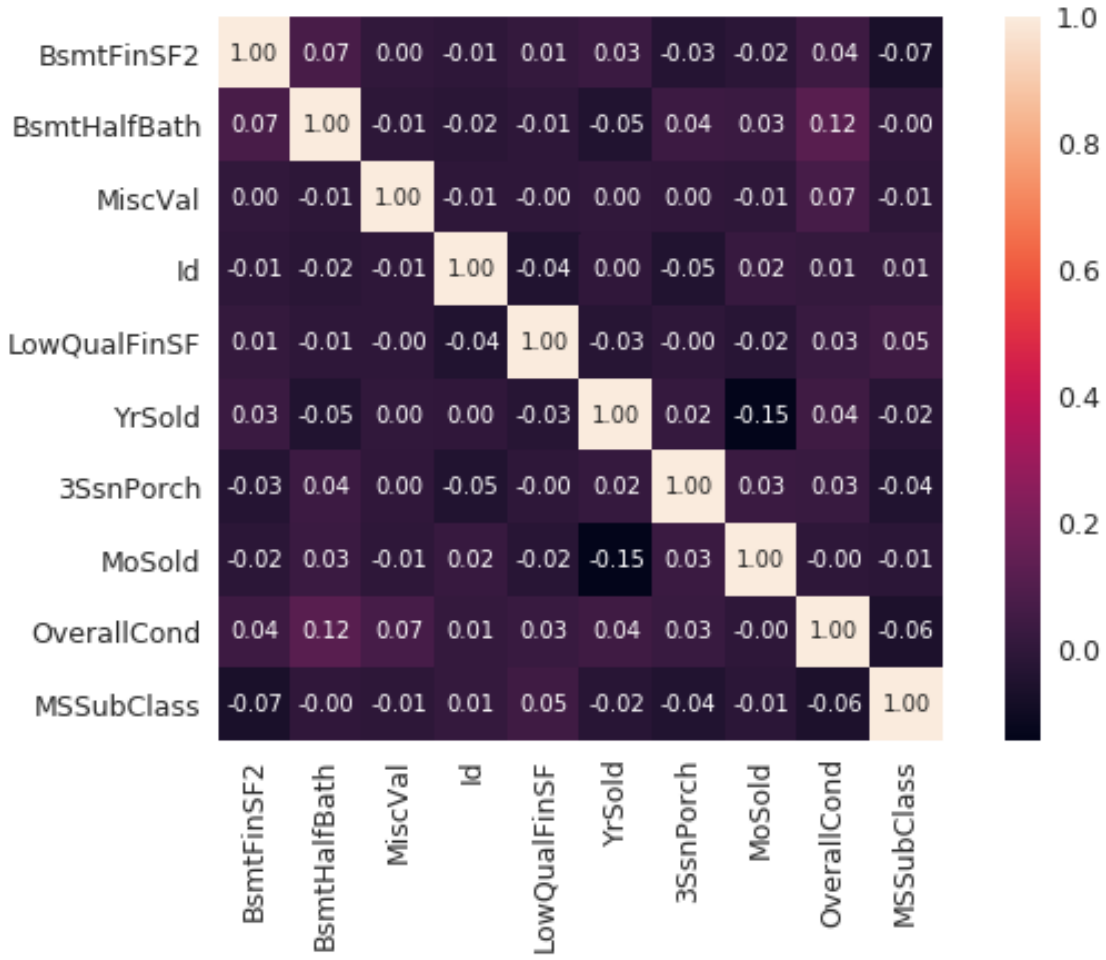
	Id	MSSubClass	LotFrontage	LotArea	OverallQual	\
BsmtFinSF2	0.005968	0.065649	0.049900	0.111170	0.059119	
BsmtHalfBath	0.020155	0.002333	0.007234	0.048046	0.040150	
MiscVal	0.006242	0.007683	0.003368	0.038068	0.031406	
Id	1.000000	0.011156	0.010601	0.033226	0.028365	
LowQualFinSF	0.044230	0.046474	0.038469	0.004779	0.030429	
YrSold	0.000712	0.021407	0.007450	0.014261	0.027347	
3SsnPorch	0.046635	0.043825	0.070029	0.020423	0.030371	
MoSold	0.021172	0.013585	0.011200	0.001205	0.070815	
OverallCond	0.012609	0.059316	0.059213	0.005636	0.091932	
MSSubClass	0.011156	1.000000	0.386347	0.139781	0.032628	

	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	\
BsmtFinSF2	0.040229	0.049107	0.067759	0.072319	0.050117	
BsmtHalfBath	0.117821	0.038162	0.012337	0.026673	0.067418	
MiscVal	0.068777	0.034383	0.010286	0.029815	0.003571	
Id	0.012609	0.012713	0.021998	0.050298	0.005024	
LowQualFinSF	0.025494	0.183784	0.062419	0.069071	0.064503	
YrSold	0.043950	0.013618	0.035743	0.008201	0.014359	
3SsnPorch	0.025504	0.031355	0.045286	0.018796	0.026451	
MoSold	0.003511	0.012398	0.021490	0.005965	0.015727	
OverallCond	1.000000	0.375983	0.073741	0.128101	0.046231	
MSSubClass	0.059316	0.027850	0.040581	0.022936	0.069836	

	...	OpenPorchSF	EnclosedPorch	3SsnPorch	ScreenPorch	\
BsmtFinSF2	...	0.003093	0.036543	0.029993	0.088871	
BsmtHalfBath	...	0.025324	0.008555	0.035114	0.032121	
MiscVal	...	0.018584	0.018361	0.000354	0.031946	

```
Id              ...       0.000477       0.002889   0.046635     0.001330
LowQualFinSF    ...       0.018251       0.061081   0.004296     0.026799
YrSold          ...       0.057619       0.009916   0.018645     0.010694
3SsnPorch       ...       0.005842       0.037305   1.000000     0.031436
MoSold          ...       0.071255       0.028887   0.029474     0.023217
OverallCond     ...       0.032589       0.070356   0.025504     0.054811
MSSubClass      ...       0.006100       0.012037   0.043825     0.026030

                PoolArea   MiscVal    MoSold    YrSold  SalePrice   TotalSF
BsmtFinSF2      0.041709  0.004940  0.015211  0.031706   0.011378  0.048916
BsmtHalfBath    0.020025  0.007367  0.032873  0.046524   0.016844  0.011921
MiscVal         0.029669  1.000000  0.006495  0.004906   0.021190  0.011186
Id              0.057044  0.006242  0.021172  0.000712   0.021917  0.000322
LowQualFinSF    0.062157  0.003793  0.022174  0.028921   0.025606  0.009207
YrSold          0.059689  0.004906  0.145721  1.000000   0.028923  0.029638
3SsnPorch       0.007992  0.000354  0.029474  0.018645   0.044584  0.033414
MoSold          0.033737  0.006495  1.000000  0.145721   0.046432  0.040485
OverallCond     0.001985  0.068777  0.003511  0.043950   0.077856  0.143814
MSSubClass      0.008283  0.007683  0.013585  0.021407   0.084284  0.082225

[10 rows x 39 columns]
```

In [28]:
```
corrmat = train_df.corr().abs()
cols = corrmat.nsmallest(10,'SalePrice')['SalePrice'].index
cm = np.corrcoef(train_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':
                 xticklabels=cols.values)
plt.yticks(rotation=0)
plt.xticks(rotation=90)
plt.show()
```




```
In [29]: corrmat = train_df.corr().abs()
         corrmat.nsmallest(30,'SalePrice')['SalePrice']
```

```
Out[29]: BsmtFinSF2       0.011378
         BsmtHalfBath     0.016844
         MiscVal          0.021190
         Id               0.021917
         LowQualFinSF     0.025606
         YrSold           0.028923
         3SsnPorch        0.044584
         MoSold           0.046432
         OverallCond      0.077856
         MSSubClass       0.084284
         PoolArea         0.092404
         ScreenPorch      0.111447
         EnclosedPorch    0.128578
         KitchenAbvGr     0.135907
```

```
BedroomAbvGr     0.168213
BsmtUnfSF        0.214479
BsmtFullBath     0.227122
LotArea          0.263843
HalfBath         0.284108
OpenPorchSF      0.315856
2ndFlrSF         0.319334
WoodDeckSF       0.324413
LotFrontage      0.351799
BsmtFinSF1       0.386420
Fireplaces       0.466929
MasVnrArea       0.477493
GarageYrBlt      0.486362
YearRemodAdd     0.507101
YearBuilt        0.522897
TotRmsAbvGrd     0.533723
Name: SalePrice, dtype: float64
```

In [30]:
```
categoricals = train_df.select_dtypes(exclude=[np.number])
categoricals.describe()
```

Out[30]:

	MSZoning	Street	Alley	LotShape	LandContour	Utilities	LotConfig	\
count	1460	1460	91	1460	1460	1460	1460	
unique	5	2	2	4	4	2	5	
top	RL	Pave	Grvl	Reg	Lvl	AllPub	Inside	
freq	1151	1454	50	925	1311	1459	1052	

	LandSlope	Neighborhood	Condition1	...	GarageType	\
count	1460	1460	1460	...	1379	
unique	3	25	9	...	6	
top	Gtl	NAmes	Norm	...	Attchd	
freq	1382	225	1260	...	870	

	GarageFinish	GarageQual	GarageCond	PavedDrive	PoolQC	Fence	\
count	1379	1379	1379	1460	7	281	
unique	3	5	5	3	3	4	
top	Unf	TA	TA	Y	Gd	MnPrv	
freq	605	1311	1326	1340	3	157	

	MiscFeature	SaleType	SaleCondition
count	54	1460	1460
unique	4	9	6
top	Shed	WD	Normal
freq	49	1267	1198

[4 rows x 43 columns]

In [31]:
```
categoricals.isnull().sum()
```

```
Out[31]: MSZoning            0
         Street              0
         Alley            1369
         LotShape            0
         LandContour         0
         Utilities           0
         LotConfig           0
         LandSlope           0
         Neighborhood        0
         Condition1          0
         Condition2          0
         BldgType            0
         HouseStyle          0
         RoofStyle           0
         RoofMatl            0
         Exterior1st         0
         Exterior2nd         0
         MasVnrType          8
         ExterQual           0
         ExterCond           0
         Foundation          0
         BsmtQual           37
         BsmtCond           37
         BsmtExposure       38
         BsmtFinType1       37
         BsmtFinType2       38
         Heating             0
         HeatingQC           0
         CentralAir          0
         Electrical          1
         KitchenQual         0
         Functional          0
         FireplaceQu       690
         GarageType         81
         GarageFinish       81
         GarageQual         81
         GarageCond         81
         PavedDrive          0
         PoolQC           1453
         Fence            1179
         MiscFeature      1406
         SaleType            0
         SaleCondition       0
         dtype: int64
```

```
In [32]: category_null = pd.concat([categoricals.isnull().sum(),categoricals.isnull().sum()/cate
         category_null[category_null.sum(axis=1)>0]
```

```
Out[32]:               category_count    %count
```

```
Alley                   1369  0.937671
MasVnrType                 8  0.005479
BsmtQual                  37  0.025342
BsmtCond                  37  0.025342
BsmtExposure              38  0.026027
BsmtFinType1              37  0.025342
BsmtFinType2              38  0.026027
Electrical                 1  0.000685
FireplaceQu              690  0.472603
GarageType                81  0.055479
GarageFinish              81  0.055479
GarageQual                81  0.055479
GarageCond                81  0.055479
PoolQC                  1453  0.995205
Fence                   1179  0.807534
MiscFeature             1406  0.963014
```

```
In [33]: categoricals.columns
```

```
Out[33]: Index(['MSZoning', 'Street', 'Alley', 'LotShape', 'LandContour', 'Utilities',
                'LotConfig', 'LandSlope', 'Neighborhood', 'Condition1', 'Condition2',
                'BldgType', 'HouseStyle', 'RoofStyle', 'RoofMatl', 'Exterior1st',
                'Exterior2nd', 'MasVnrType', 'ExterQual', 'ExterCond', 'Foundation',
                'BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2',
                'Heating', 'HeatingQC', 'CentralAir', 'Electrical', 'KitchenQual',
                'Functional', 'FireplaceQu', 'GarageType', 'GarageFinish', 'GarageQual',
                'GarageCond', 'PavedDrive', 'PoolQC', 'Fence', 'MiscFeature',
                'SaleType', 'SaleCondition'],
               dtype='object')
```

```
In [34]: train_df[ 'RoofMatl'].describe()
```

```
Out[34]: count        1460
         unique          8
         top       CompShg
         freq         1434
         Name: RoofMatl, dtype: object
```

```
In [35]: roof = pd.get_dummies(train_df.RoofMatl, drop_first=True)
         roof.describe()
```

```
Out[35]:            CompShg      Membran        Metal         Roll      Tar&Grv  \
         count  1460.000000  1460.000000  1460.000000  1460.000000  1460.000000
         mean      0.982192     0.000685     0.000685     0.000685     0.007534
         std       0.132299     0.026171     0.026171     0.026171     0.086502
         min       0.000000     0.000000     0.000000     0.000000     0.000000
         25%       1.000000     0.000000     0.000000     0.000000     0.000000
         50%       1.000000     0.000000     0.000000     0.000000     0.000000
         75%       1.000000     0.000000     0.000000     0.000000     0.000000
```

```
max       1.000000     1.000000     1.000000     1.000000     1.000000

           WdShake      WdShngl
count  1460.000000  1460.000000
mean      0.003425     0.004110
std       0.058440     0.063996
min       0.000000     0.000000
25%       0.000000     0.000000
50%       0.000000     0.000000
75%       0.000000     0.000000
max       1.000000     1.000000
```

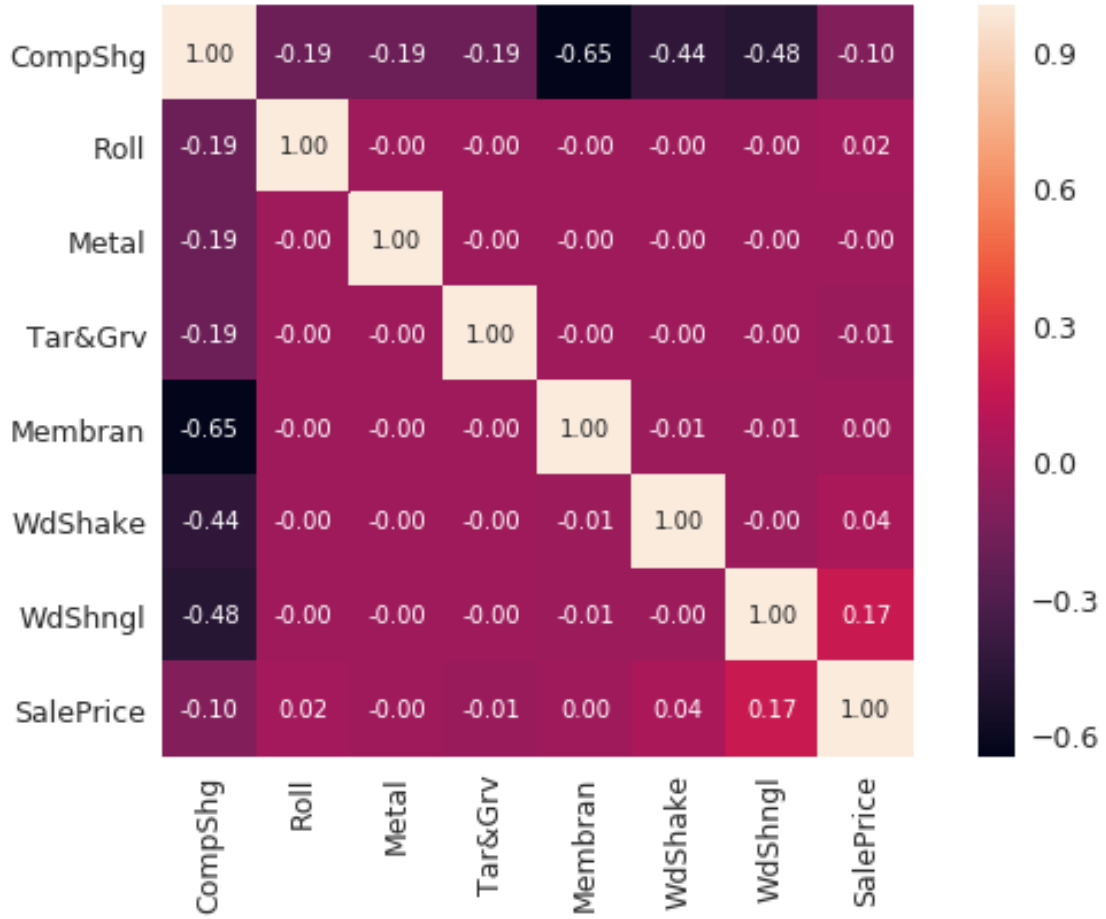
In [36]:
```
df = pd.concat([roof,train_df['SalePrice']],axis=1)
```

In [37]:
```
df.head()
```

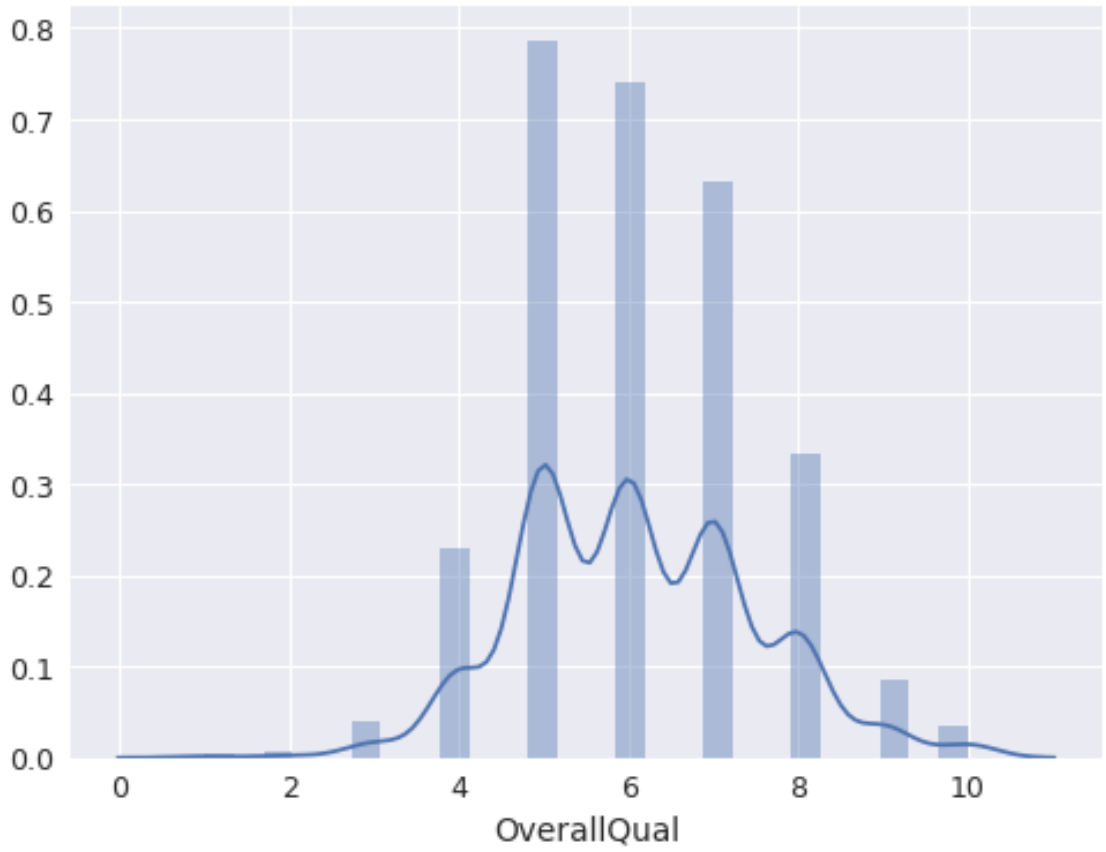
Out[37]:

	CompShg	Membran	Metal	Roll	Tar&Grv	WdShake	WdShngl	SalePrice
0	1	0	0	0	0	0	0	208500
1	1	0	0	0	0	0	0	181500
2	1	0	0	0	0	0	0	223500
3	1	0	0	0	0	0	0	140000
4	1	0	0	0	0	0	0	250000

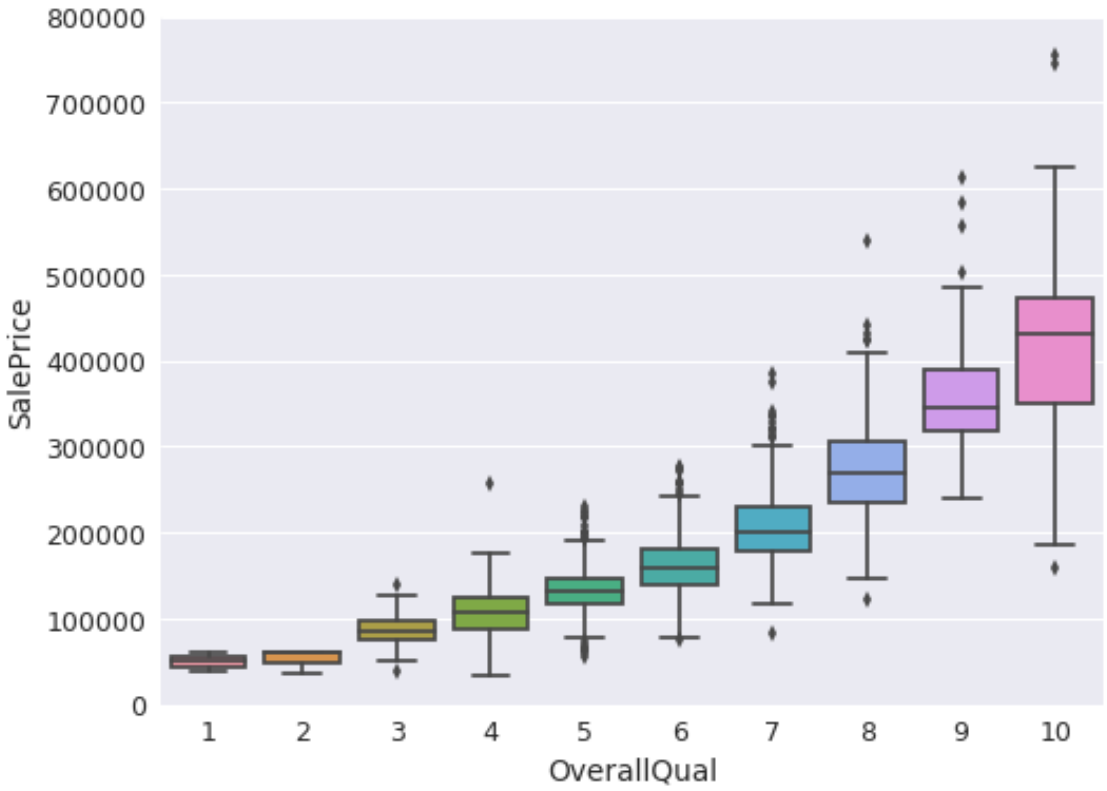
In [38]:
```
corrmat = df.corr()
cols = corrmat.nsmallest(8,'SalePrice')['SalePrice'].index
cm = np.corrcoef(df[cols].values.T)
sns.set(font_scale=1.25)
hm = sns.heatmap(corrmat, cbar=True, annot=True, square=True, fmt='.2f', annot_kws={'si
                 xticklabels=cols.values)
plt.yticks(rotation=0)
plt.xticks(rotation=90)
plt.show()
```



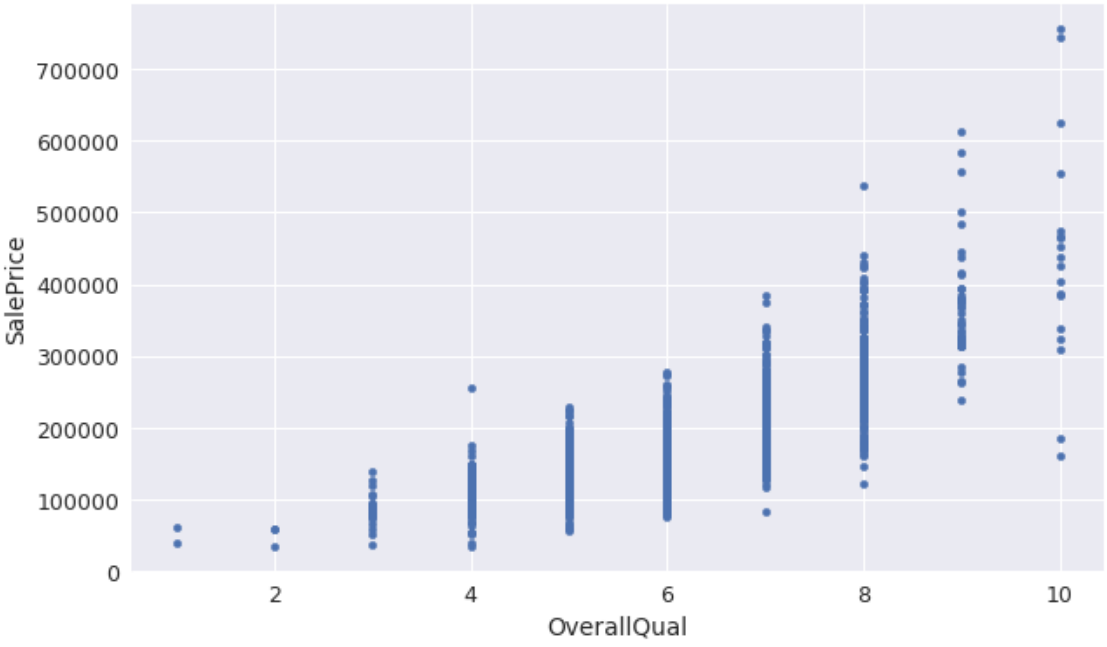

```
In [39]: train_df['OverallQual'].describe()

Out[39]: count    1460.000000
         mean        6.099315
         std         1.382997
         min         1.000000
         25%         5.000000
         50%         6.000000
         75%         7.000000
         max        10.000000
         Name: OverallQual, dtype: float64

In [40]: f, ax = plt.subplots(figsize=(8, 6))
         fig = sns.distplot(train_df['OverallQual'])
         plt.show()
```



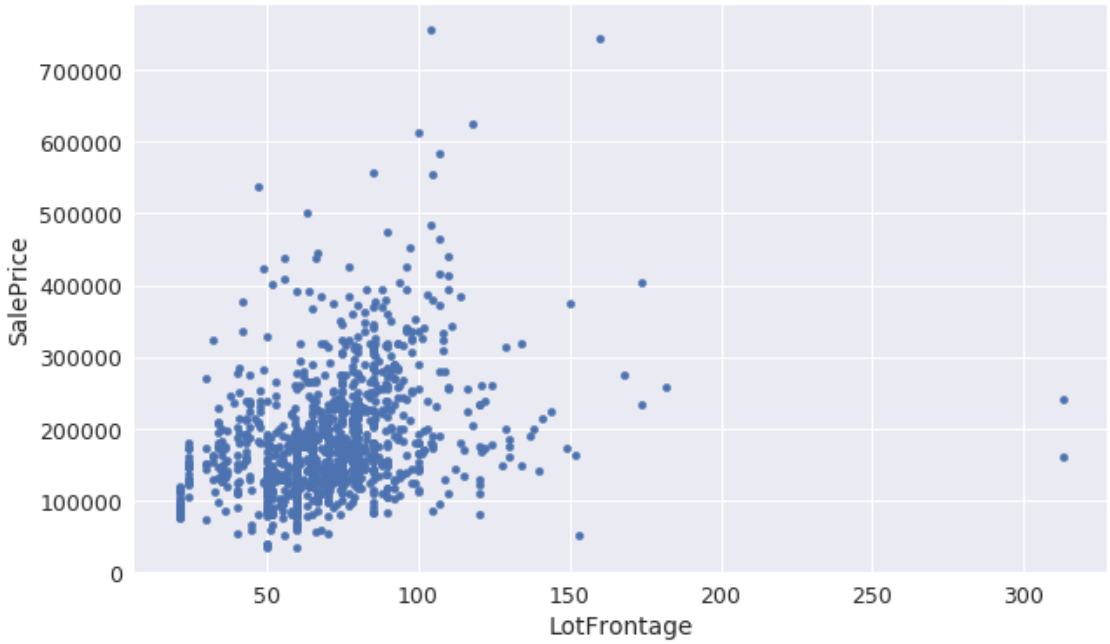

```
In [41]: overall_qual = pd.concat([train_df['SalePrice'], train_df['OverallQual']], axis=1)
         f, ax = plt.subplots(figsize=(8, 6))
         fig = sns.boxplot(x='OverallQual', y="SalePrice", data=overall_qual)
         fig.axis(ymin=0, ymax=800000)
         plt.show()
```



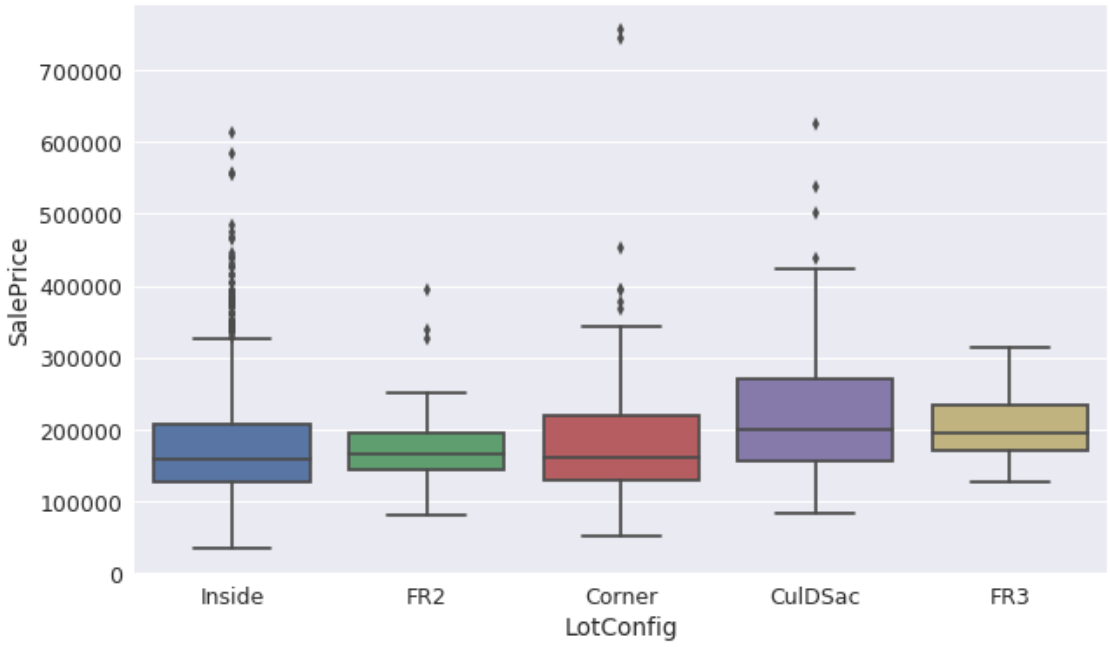
In [42]:
```
overall_qual.plot.scatter(x='OverallQual', y="SalePrice")
plt.show()
```



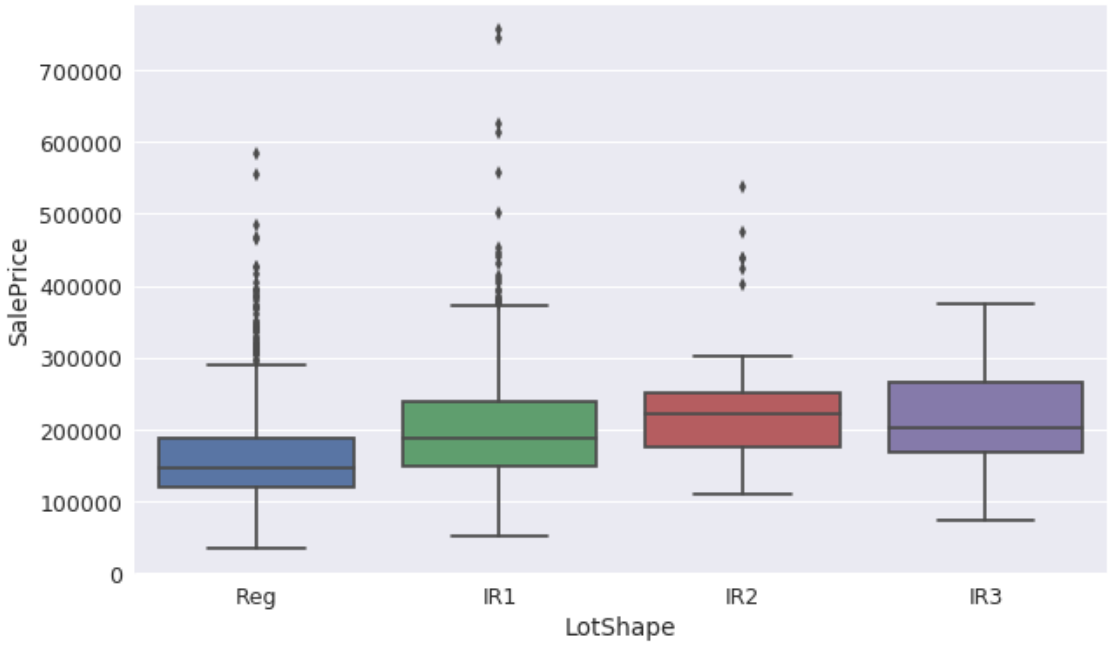
```
In [43]: data = pd.concat([train_df['SalePrice'],train_df['LotFrontage']],axis=1)
         data.plot.scatter(x='LotFrontage', y="SalePrice")
         plt.show()
```




```
In [44]: train_df['LotConfig'].describe()

Out[44]: count       1460
         unique         5
         top       Inside
         freq        1052
         Name: LotConfig, dtype: object

In [45]: data = pd.concat([train_df['SalePrice'],train_df['LotConfig']],axis=1)
         fig = sns.boxplot(x='LotConfig', y="SalePrice", data=data)
         plt.show()
```

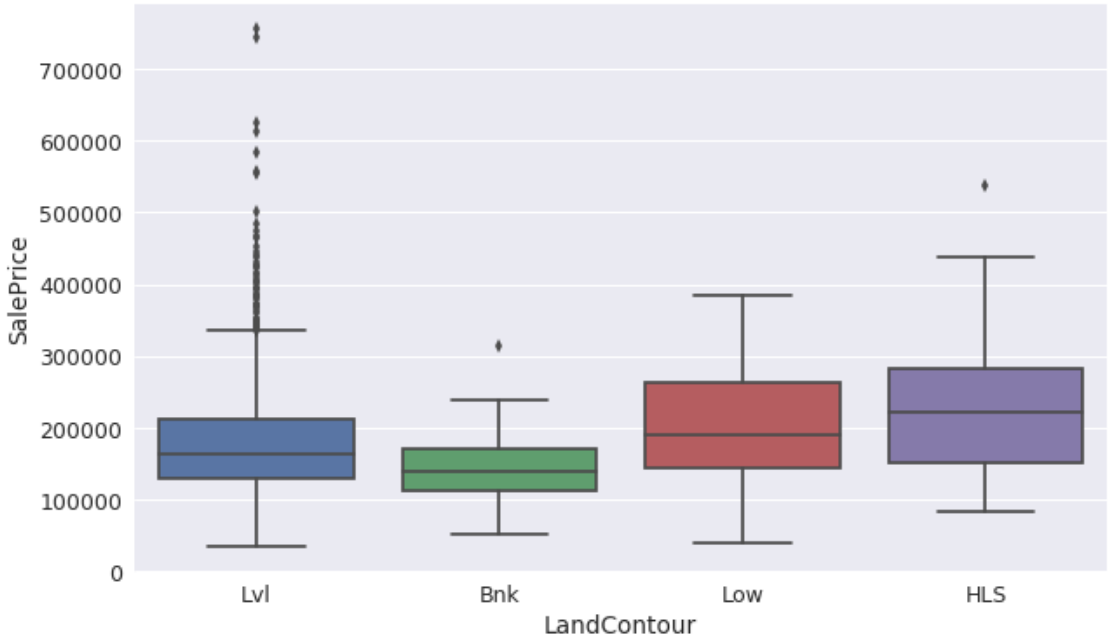



```
In [46]: train_df['LotShape'].describe()

Out[46]: count     1460
         unique       4
         top        Reg
         freq       925
         Name: LotShape, dtype: object

In [47]: data = pd.concat([train_df['SalePrice'],train_df['LotShape']],axis=1)
         fig = sns.boxplot(x='LotShape', y="SalePrice", data=data)
         plt.show()
```

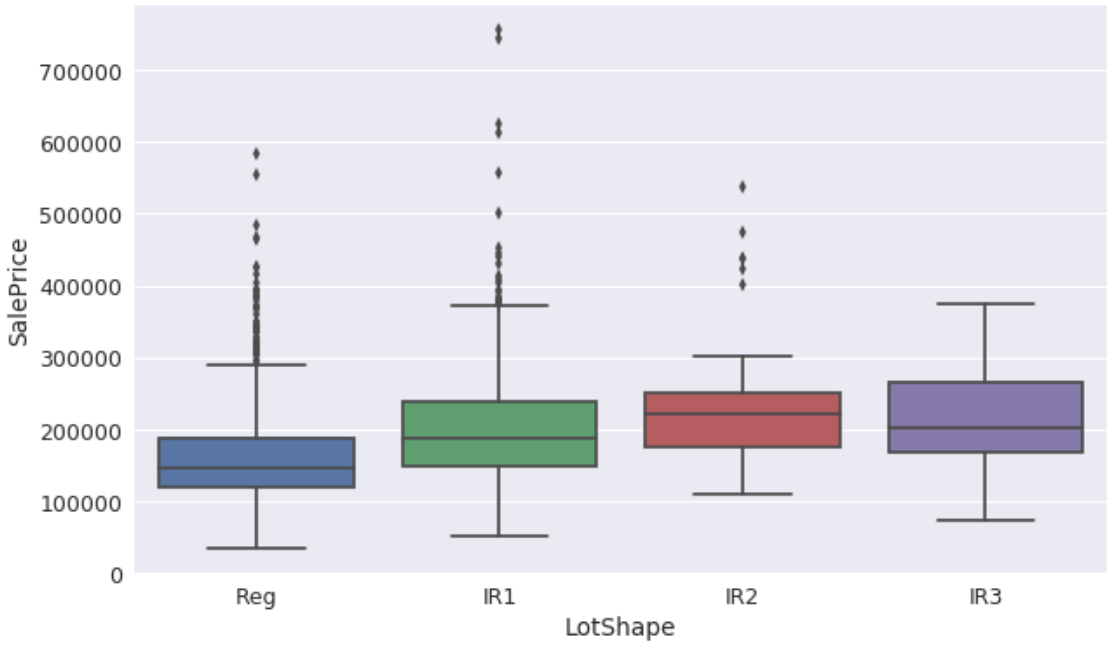


In [48]:
```
data = pd.concat([train_df['SalePrice'],train_df['LotArea']],axis=1)
data.plot.scatter(x='LotArea', y="SalePrice")
plt.show()
```

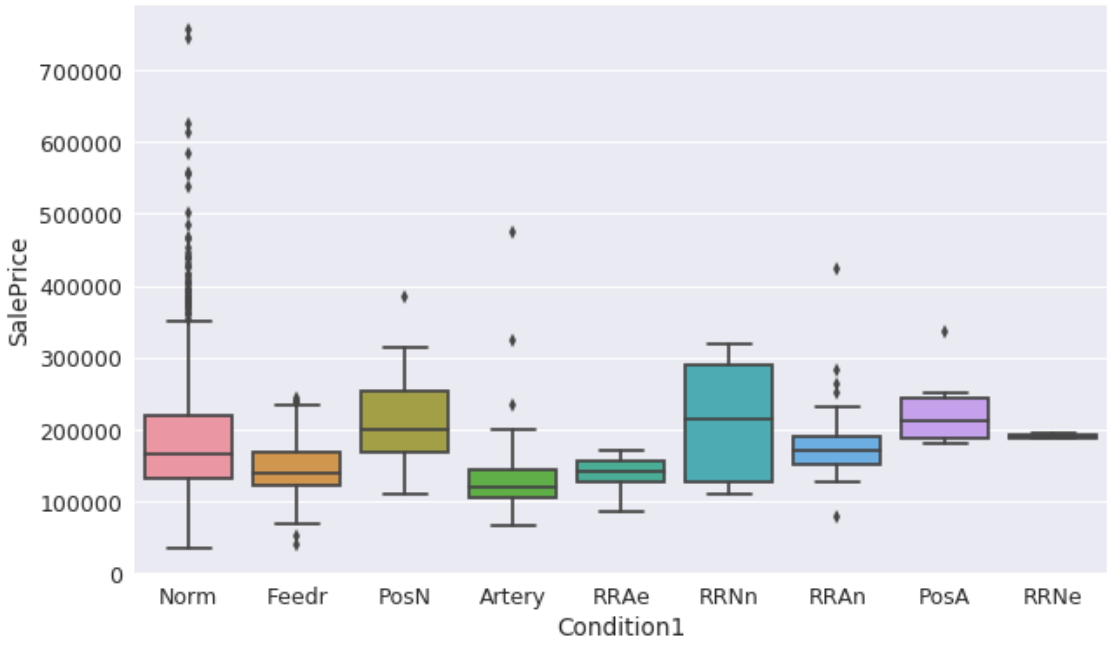
```
In [49]: train_df['LandContour'].describe()

Out[49]: count     1460
         unique       4
         top        Lvl
         freq      1311
         Name: LandContour, dtype: object

In [50]: data = pd.concat([train_df['LandContour'],train_df['SalePrice']],axis=1)
         fig = sns.boxplot(x='LandContour',y='SalePrice',data=data)
         plt.show()
```



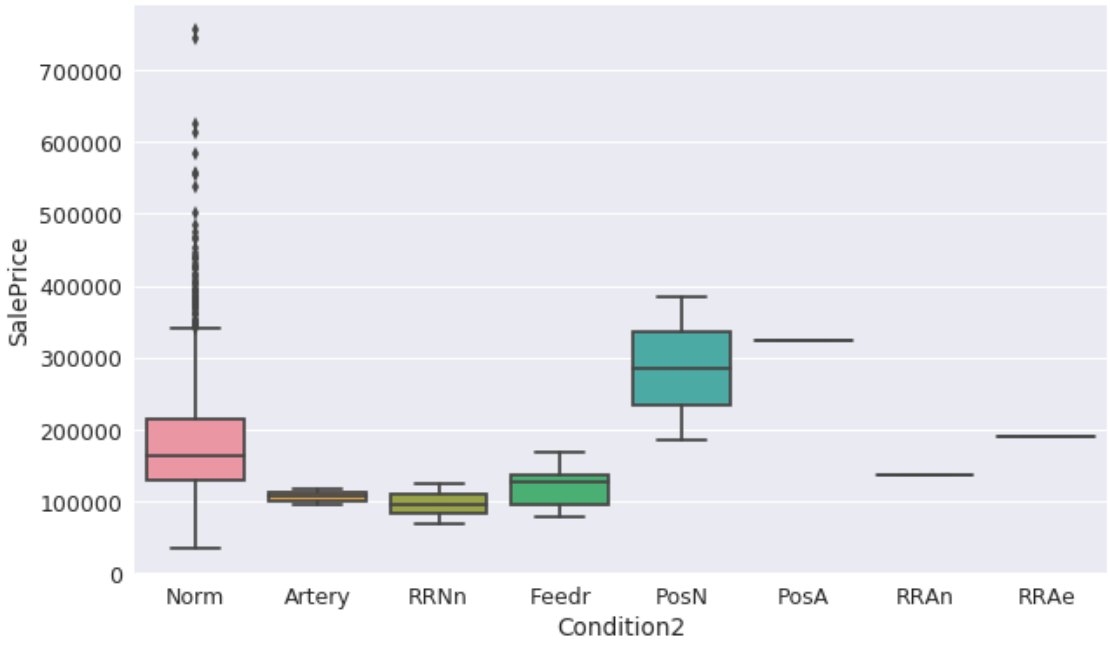

```
In [51]: train_df['LotShape'].describe()

Out[51]: count     1460
         unique       4
         top        Reg
         freq       925
         Name: LotShape, dtype: object

In [52]: data = pd.concat([train_df['SalePrice'],train_df['LotShape']],axis=1)
         fig = sns.boxplot(x='LotShape', y="SalePrice",data=data)
         plt.show()
```



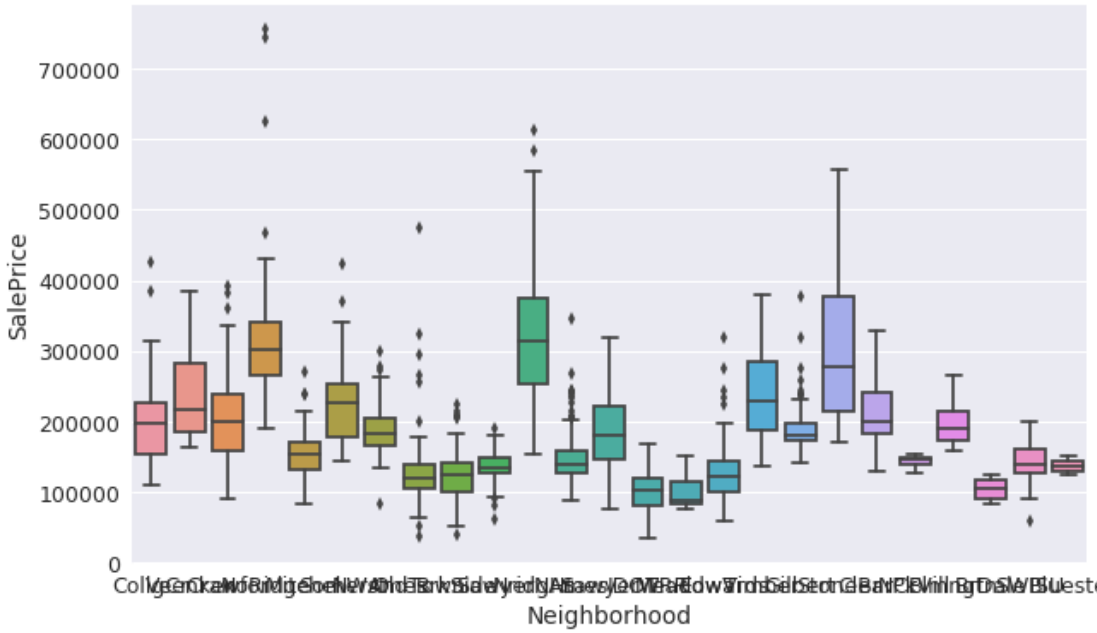

```
In [53]: train_df['Condition1'].describe()

Out[53]: count     1460
         unique       9
         top       Norm
         freq      1260
         Name: Condition1, dtype: object

In [54]: data = pd.concat([train_df['SalePrice'],train_df['Condition1']],axis=1)
         fig = sns.boxplot(x='Condition1', y="SalePrice",data=data)
         plt.show()
```



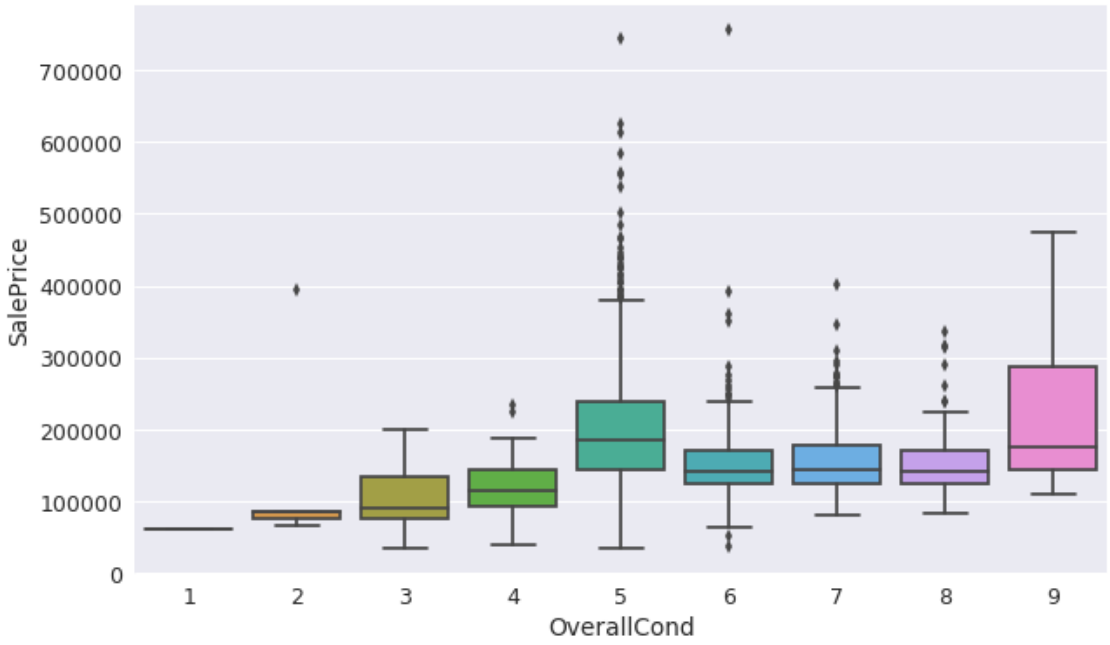

```
In [55]: train_df['Condition2'].describe()

Out[55]: count     1460
         unique       8
         top       Norm
         freq      1445
         Name: Condition2, dtype: object

In [56]: data = pd.concat([train_df['SalePrice'],train_df['Condition2']],axis=1)
         fig = sns.boxplot(x='Condition2', y="SalePrice",data=data)
         plt.show()
```



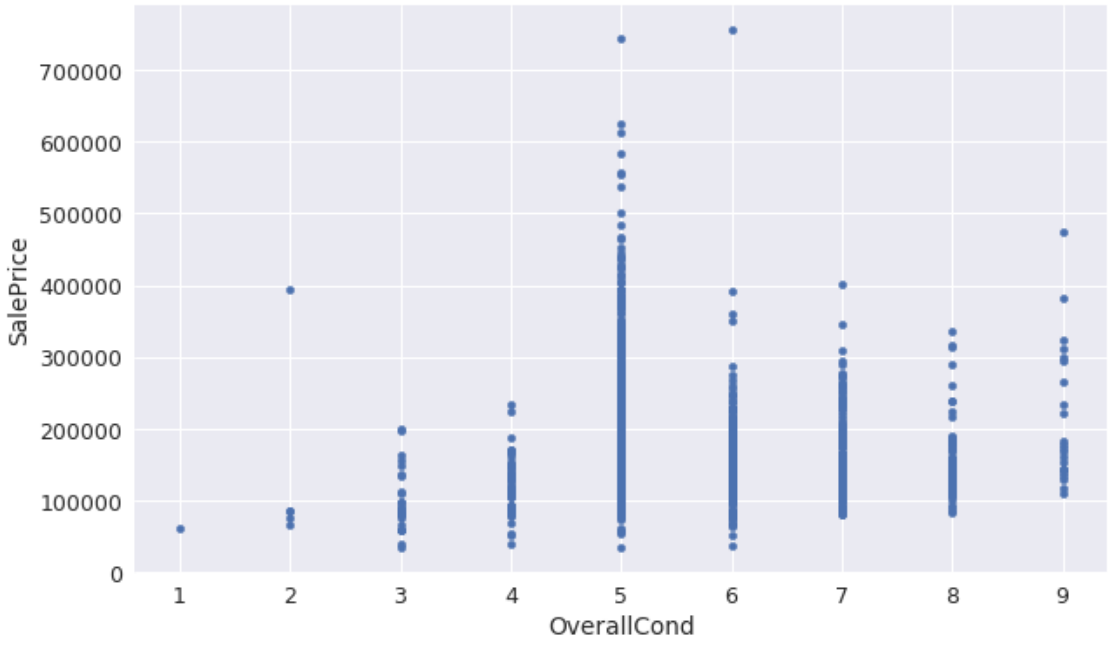

```
In [57]: train_df['Neighborhood'].describe()

Out[57]: count      1460
         unique       25
         top       NAmes
         freq        225
         Name: Neighborhood, dtype: object

In [58]: data = pd.concat([train_df['SalePrice'],train_df['Neighborhood']],axis=1)
         fig = sns.boxplot(x='Neighborhood', y="SalePrice",data=data)
         plt.show()
```



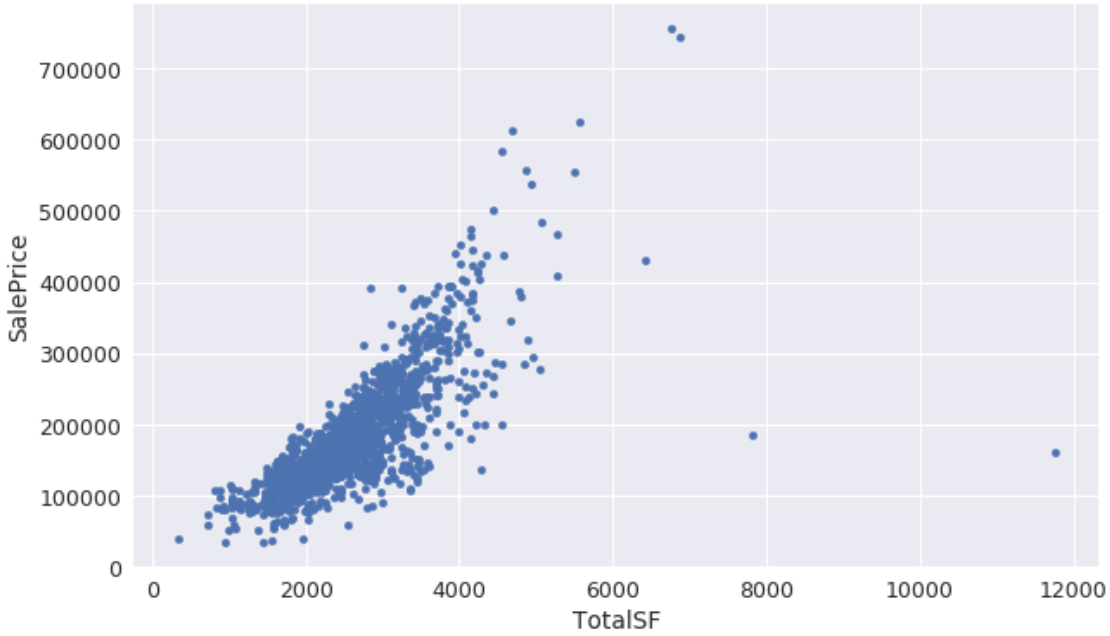

```
In [59]: train_df['OverallCond'].describe()

Out[59]: count    1460.000000
         mean        5.575342
         std         1.112799
         min         1.000000
         25%         5.000000
         50%         5.000000
         75%         6.000000
         max         9.000000
         Name: OverallCond, dtype: float64

In [60]: data = pd.concat([train_df['SalePrice'],train_df['OverallCond']],axis=1)
         fig = sns.boxplot(x='OverallCond', y="SalePrice",data=data)
         plt.show()
```



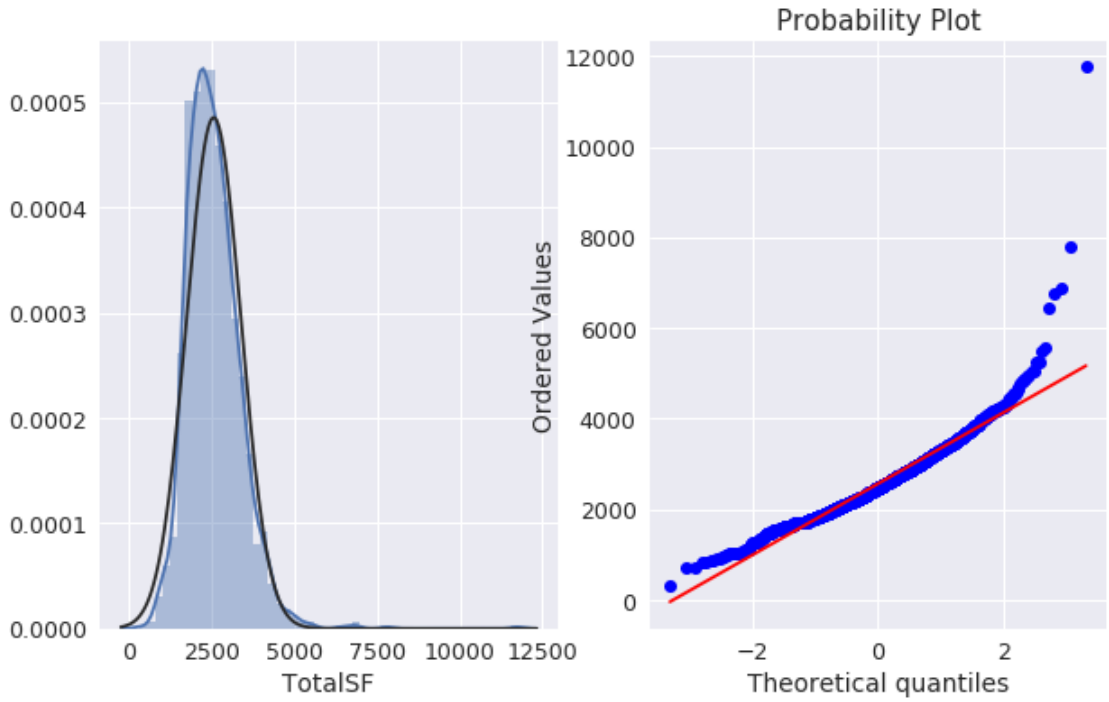

```
In [61]: data.plot.scatter(x='OverallCond', y="SalePrice")
         plt.show()
```



In [62]:
```
data = pd.concat([train_df['SalePrice'],train_df['TotalSF']],axis=1)
data.plot.scatter(x='TotalSF', y="SalePrice")
plt.show()
```



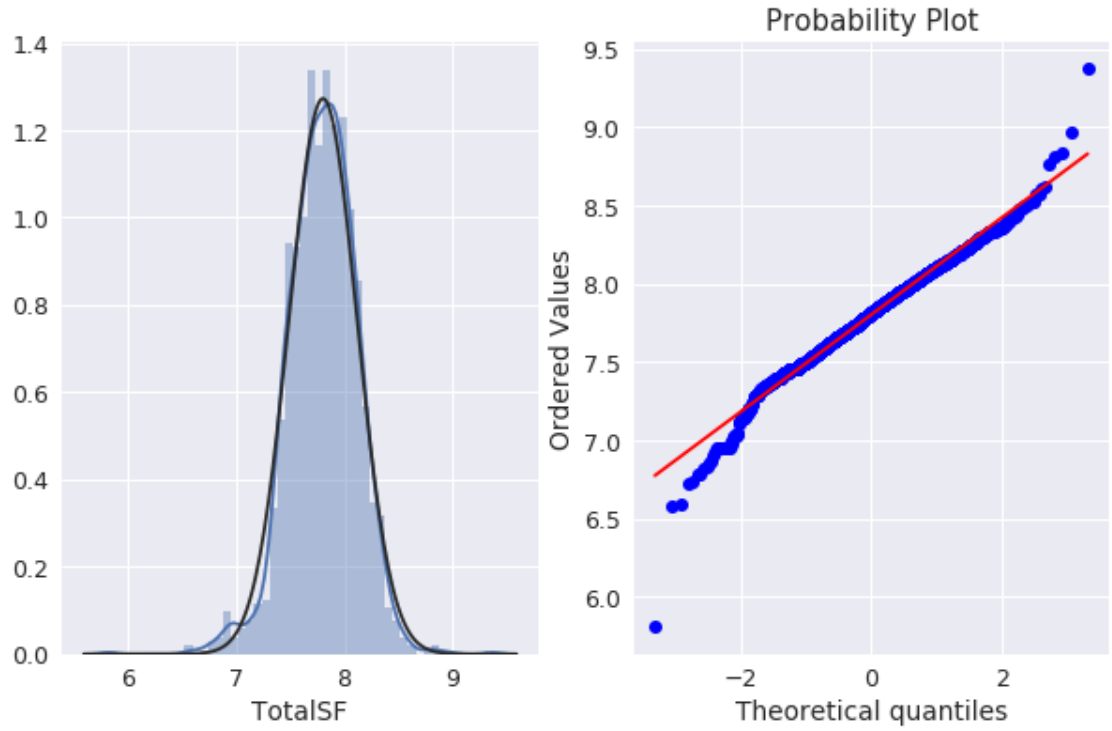
In [63]:
```
plt.figure();
plt.subplot(1,2,1)
sns.distplot(train_df['TotalSF'],fit=stats.norm)
plt.subplot(1,2,2)
stats.probplot(train_df['TotalSF'],plot=plt)
plt.show()
```




```
In [64]: print(train_df['TotalSF'].skew())
         print(train_df['TotalSF'].kurt())
```

```
1.77669962724
12.6219682969
```

```
In [65]: plt.figure();
         plt.subplot(1,2,1)
         sns.distplot(np.log(train_df['TotalSF']),fit=stats.norm)
         plt.subplot(1,2,2)
         stats.probplot(np.log(train_df['TotalSF']),plot=plt)
         plt.show()
```



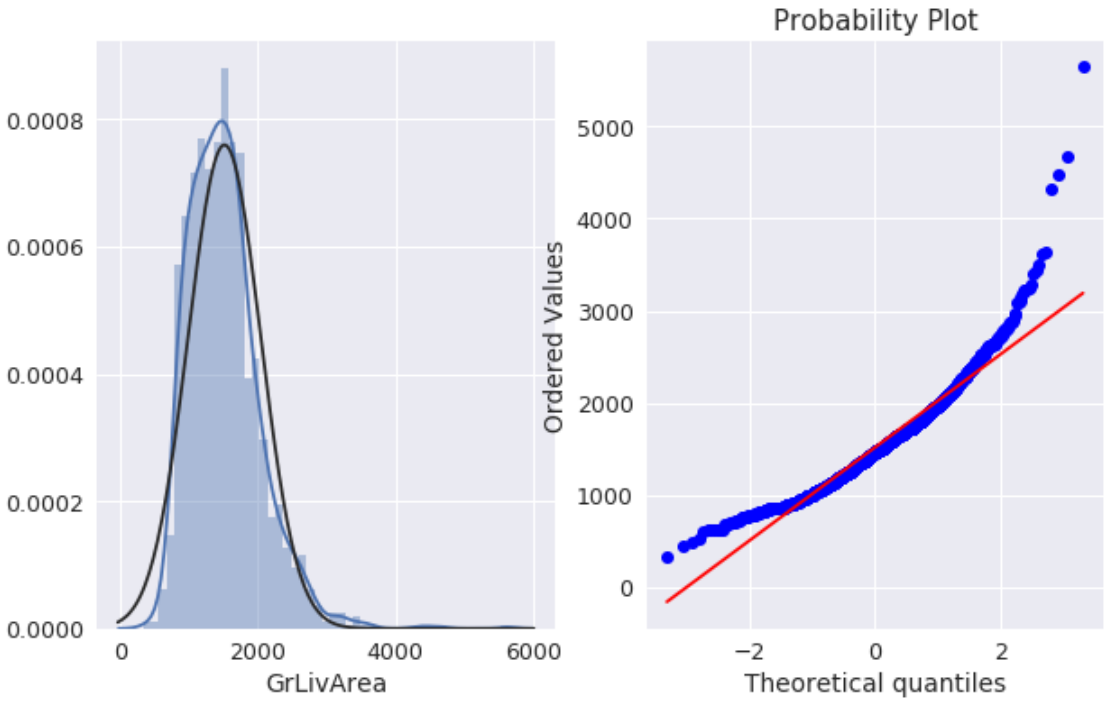

```
In [66]: print(np.log(train_df['TotalSF'].skew()))
```

```
0.574757501254
```

```
In [67]: print(np.log(train_df['TotalSF'].kurt()))
```

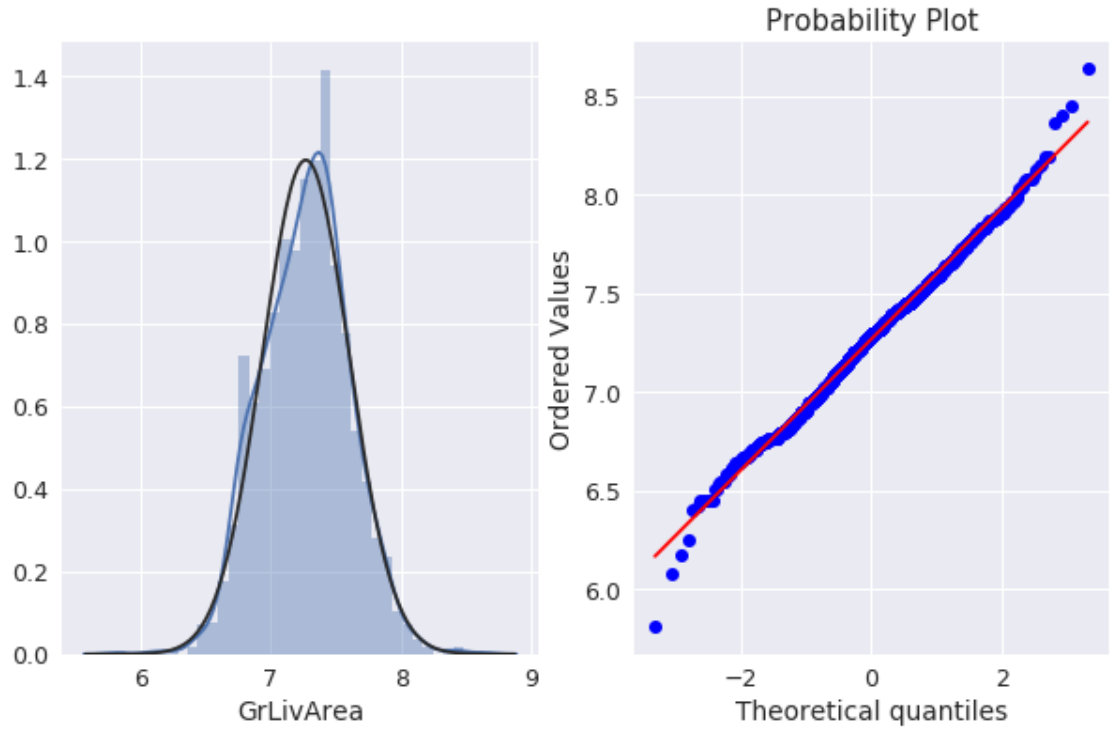
```
2.53543881142
```

```
In [68]: plt.figure();
         plt.subplot(1,2,1)
         sns.distplot(train_df['GrLivArea'],fit=stats.norm)
         plt.subplot(1,2,2)
         stats.probplot(train_df['GrLivArea'],plot=plt)
         plt.show()
         print(train_df['GrLivArea'].skew())
         print(train_df['GrLivArea'].kurt())
```




```
1.36656035602
4.89512058069
```

```
In [69]: plt.figure();
         plt.subplot(1,2,1)
         sns.distplot(np.log(train_df['GrLivArea']),fit=stats.norm)
         plt.subplot(1,2,2)
         stats.probplot(np.log(train_df['GrLivArea']),plot=plt)
         plt.show()
         print(np.log(train_df['GrLivArea']).skew())
         print(np.log(train_df['GrLivArea']).kurt())
```

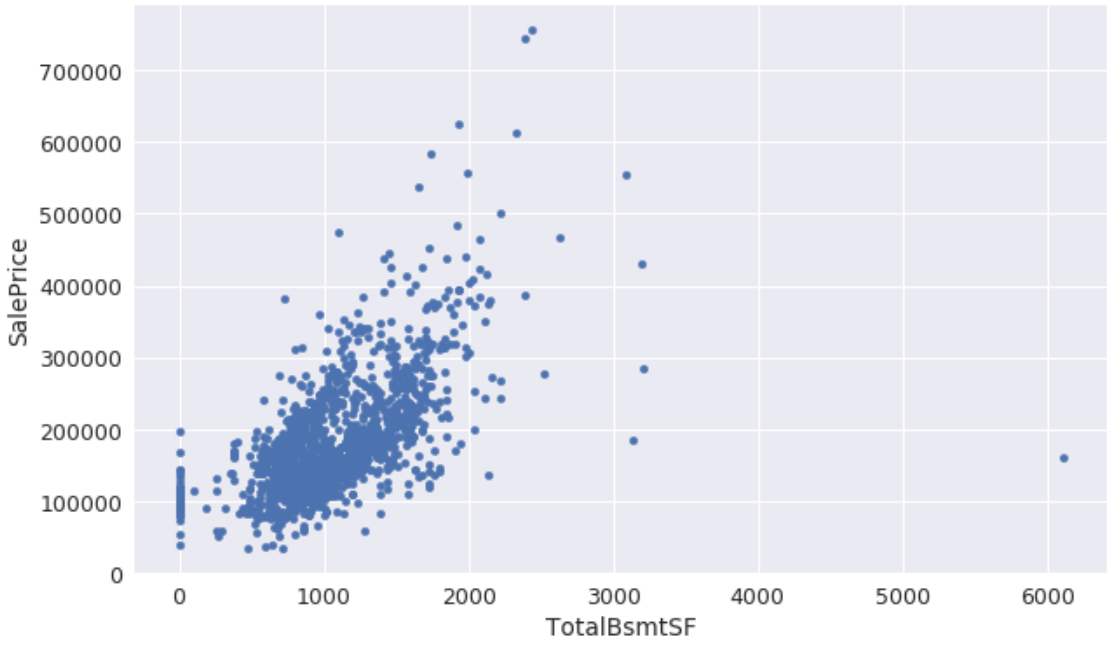



```
-0.00699518218769
0.282602688501
```

```
In [70]: train_df['TotalBsmtSF'].describe()
```

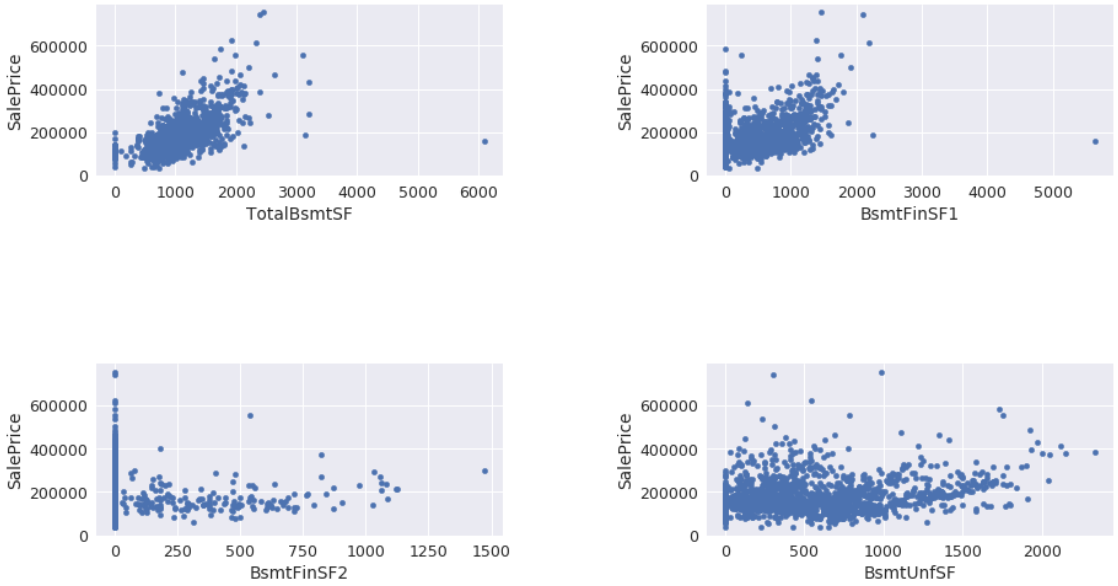
```
Out[70]: count    1460.000000
         mean     1057.429452
         std       438.705324
         min         0.000000
         25%       795.750000
         50%       991.500000
         75%      1298.250000
         max      6110.000000
         Name: TotalBsmtSF, dtype: float64
```

```
In [71]: data1 = pd.concat([train_df['TotalBsmtSF'],train_df['SalePrice']],axis=1)
         data1.plot.scatter(x='TotalBsmtSF',y='SalePrice')
         plt.show()
```



In [72]:
```
f, ax1 = plt.subplots(figsize=(20,15) ,ncols=2, nrows=2)

left   =  0.125  # the left side of the subplots of the figure
right  =  0.7    # the right side of the subplots of the figure
bottom =  0.1    # the bottom of the subplots of the figure
top    =  0.5    # the top of the subplots of the figure
wspace =  .5     # the amount of width reserved for blank space between subplots
hspace =  1.1    # the amount of height reserved for white space between subplots


plt.subplots_adjust(
    left    =  left,
    bottom  =  bottom,
    right   =  right,
    top     =  top,
    wspace  =  wspace,
    hspace  =  hspace
)
data1 = pd.concat([train_df['TotalBsmtSF'],train_df['SalePrice']],axis=1)
data1.plot.scatter(x='TotalBsmtSF',y='SalePrice',ax=ax1[0][0])
data1 = pd.concat([train_df['BsmtFinSF1'],train_df['SalePrice']],axis=1)
data1.plot.scatter(x='BsmtFinSF1',y='SalePrice',ax=ax1[0][1])
data1 = pd.concat([train_df['BsmtFinSF2'],train_df['SalePrice']],axis=1)
data1.plot.scatter(x='BsmtFinSF2',y='SalePrice',ax=ax1[1][0])
data1 = pd.concat([train_df['BsmtUnfSF'],train_df['SalePrice']],axis=1)
```

```
data1.plot.scatter(x='BsmtUnfSF',y='SalePrice',ax=ax1[1][1])
plt.show()
```




```
In [73]: train_df['Exterior2nd'].describe()

Out[73]: count         1460
         unique          16
         top        VinylSd
         freq           504
         Name: Exterior2nd, dtype: object

In [74]: train_df['Exterior2nd'].mode()

Out[74]: 0    VinylSd
         dtype: object

In [75]: train_df['MSSubClass'].describe()

Out[75]: count    1460.000000
         mean       56.897260
         std        42.300571
         min        20.000000
         25%        20.000000
         50%        50.000000
         75%        70.000000
         max       190.000000
         Name: MSSubClass, dtype: float64

In [76]: train_df['MSSubClass'].head()
```

```
Out[76]: 0    60
         1    20
         2    60
         3    70
         4    60
         Name: MSSubClass, dtype: int64
```

```
In [77]: train_df1 = pd.get_dummies(train_df,drop_first=True)
```

```
In [78]: train_df1.head()
```

```
Out[78]:    Id  MSSubClass  LotFrontage  LotArea  OverallQual  OverallCond  YearBuilt  \
         0   1          60         65.0     8450            7            5       2003
         1   2          20         80.0     9600            6            8       1976
         2   3          60         68.0    11250            7            5       2001
         3   4          70         60.0     9550            7            5       1915
         4   5          60         84.0    14260            8            5       2000

            YearRemodAdd  MasVnrArea  BsmtFinSF1          ...            \
         0          2003       196.0         706          ...
         1          1976         0.0         978          ...
         2          2002       162.0         486          ...
         3          1970         0.0         216          ...
         4          2000       350.0         655          ...

            SaleType_ConLI  SaleType_ConLw  SaleType_New  SaleType_Oth  SaleType_WD  \
         0               0               0             0             0            1
         1               0               0             0             0            1
         2               0               0             0             0            1
         3               0               0             0             0            1
         4               0               0             0             0            1

            SaleCondition_AdjLand  SaleCondition_Alloca  SaleCondition_Family  \
         0                      0                     0                     0
         1                      0                     0                     0
         2                      0                     0                     0
         3                      0                     0                     0
         4                      0                     0                     0

            SaleCondition_Normal  SaleCondition_Partial
         0                     1                     0
         1                     1                     0
         2                     1                     0
         3                     0                     0
         4                     1                     0

         [5 rows x 248 columns]
```

```
In [79]: ## After this we do variable dropping and variable manipulation for linear regression
```

```
train = pd.read_csv('../input/train.csv')
test = pd.read_csv('../input/test.csv')
```

In [80]:
```
train.drop(['Id','Alley','FireplaceQu', 'PoolQC','Fence','MiscFeature'],axis=1,inplace=
```

In [81]:
```
train.drop(['BsmtFinSF1', 'BsmtFinSF2', 'BsmtUnfSF'], axis=1, inplace=True)
```

In [82]:
```
train['TotalBsmtSF'] = train['TotalBsmtSF'].fillna(0)
train['1stFlrSF'] = train['1stFlrSF'].fillna(0)
train['2ndFlrSF'] = train['2ndFlrSF'].fillna(0)
train['TotalSF'] = train['TotalBsmtSF'] + train['1stFlrSF'] + train['2ndFlrSF']
train.drop(['TotalBsmtSF', '1stFlrSF', '2ndFlrSF'], axis=1, inplace=True)
train.drop(['GarageArea','TotRmsAbvGrd'], axis=1, inplace=True) # as analysis before
```

In [83]:
```
train.drop(['Utilities', 'RoofMatl', 'MasVnrArea', 'MasVnrType', 'Heating', 'LowQualFin
            'BsmtFullBath', 'BsmtHalfBath', 'BsmtQual', 'BsmtCond', 'BsmtExposure', 'Bs
            'Functional', 'GarageYrBlt', 'GarageCond', 'GarageType', 'GarageFinish', 'G
            'OpenPorchSF', 'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea',
            'MiscVal'], axis=1, inplace=True)
```

In [84]:
```
## Normalize data
numeric_data = train.loc[:, ['LotFrontage', 'LotArea', 'GrLivArea', 'TotalSF']]
numeric_data_standardized = (numeric_data - numeric_data.mean())/numeric_data.std()
```

In [85]:
```
#  Fillling nan values
train['MSZoning'] = train['MSZoning'].fillna(train['MSZoning'].mode()[0])
train['LotFrontage'] = train['LotFrontage'].fillna(train['LotFrontage'].mean())
train['Electrical'] = train['Electrical'].fillna(train['Electrical'].mode()[0])
train['KitchenQual'] = train['KitchenQual'].fillna(train['KitchenQual'].mode()[0])
train['GarageCars'] = train['GarageCars'].fillna(0.0)
train['SaleType'] = train['SaleType'].fillna(train['SaleType'].mode()[0])
train['Exterior1st'] = train['Exterior1st'].fillna(train['Exterior2nd'].mode()[0])
train['Exterior2nd'] = train['Exterior2nd'].fillna(train['Exterior2nd'].mode()[0])
```

In [86]:
```
train['MSSubClass'] = train['MSSubClass'].astype(str)
train['OverallCond'] = train['OverallCond'].astype(str)
train['KitchenAbvGr'] = train['KitchenAbvGr'].astype(str)

train['YrSold'] = train['YrSold'].astype(str)
train['MoSold'] = train['MoSold'].astype(str)
```

In [87]:
```
def encode(x): return 1 if x == 'Partial' else 0
train['enc_condition'] = train.SaleCondition.apply(encode)
train.drop(['SaleType'],axis=1,inplace=True)
```

In [88]:
```
train.isnull().sum()
```

Out[88]:
```
MSSubClass       0
MSZoning         0
```

```
LotFrontage      0
LotArea          0
Street           0
LotShape         0
LandContour      0
LotConfig        0
LandSlope        0
Neighborhood     0
Condition1       0
Condition2       0
BldgType         0
HouseStyle       0
OverallQual      0
OverallCond      0
YearBuilt        0
YearRemodAdd     0
RoofStyle        0
Exterior1st      0
Exterior2nd      0
ExterQual        0
ExterCond        0
Foundation       0
HeatingQC        0
CentralAir       0
Electrical       0
GrLivArea        0
FullBath         0
HalfBath         0
BedroomAbvGr     0
KitchenAbvGr     0
KitchenQual      0
Fireplaces       0
GarageCars       0
PavedDrive       0
MoSold           0
YrSold           0
SaleCondition    0
SalePrice        0
TotalSF          0
enc_condition    0
dtype: int64
```

```
In [89]: sum(train.isnull().sum() != 0)
```

```
Out[89]: 0
```

```
In [90]: print(train.shape[0])
         print(train.shape[1])
```

```
1460
42
```

```
In [91]: train.head()
```

Out[91]:

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

	LotConfig	LandSlope	Neighborhood	...	KitchenQual	Fireplaces	\
0	Inside	Gtl	CollgCr	...	Gd	0	
1	FR2	Gtl	Veenker	...	TA	1	
2	Inside	Gtl	CollgCr	...	Gd	1	
3	Corner	Gtl	Crawfor	...	Gd	1	
4	FR2	Gtl	NoRidge	...	Gd	1	

	GarageCars	PavedDrive	MoSold	YrSold	SaleCondition	SalePrice	TotalSF	\
0	2	Y	2	2008	Normal	208500	2566	
1	2	Y	5	2007	Normal	181500	2524	
2	2	Y	9	2008	Normal	223500	2706	
3	3	Y	2	2006	Abnorml	140000	2473	
4	3	Y	12	2008	Normal	250000	3343	

	enc_condition
0	0
1	0
2	0
3	0
4	0

[5 rows x 42 columns]

```
In [92]: y = np.log(train.SalePrice)
         X = train

In [93]: X.drop(['SalePrice'],axis=1,inplace=True)

In [94]: print(y.shape[0])
         print(X.shape[0])
         print(X.shape[1])

1460
1460
41
```

```
In [95]: X.dtypes.sample(20)
```

```
Out[95]: Foundation       object
         YearBuilt         int64
         Neighborhood     object
         GarageCars        int64
         KitchenAbvGr     object
         GrLivArea         int64
         MoSold           object
         LotArea           int64
         Exterior1st      object
         SaleCondition    object
         BedroomAbvGr      int64
         LandContour      object
         Electrical       object
         Fireplaces        int64
         Exterior2nd      object
         Condition2       object
         Street           object
         ExterCond        object
         enc_condition     int64
         YearRemodAdd      int64
         dtype: object
```

```
In [96]: X1_hot_code = pd.get_dummies(X,drop_first=True)
```

```
In [97]: from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(
                                                X1_hot_code, y, random_state=42, test_size=.33)
         from sklearn import linear_model
         lr = linear_model.LinearRegression()

         model = lr.fit(X_train, y_train)
```

```
In [98]: print ("R^2 is: n", model.score(X_test, y_test))
```

```
R^2 is: n 0.884519057669
```

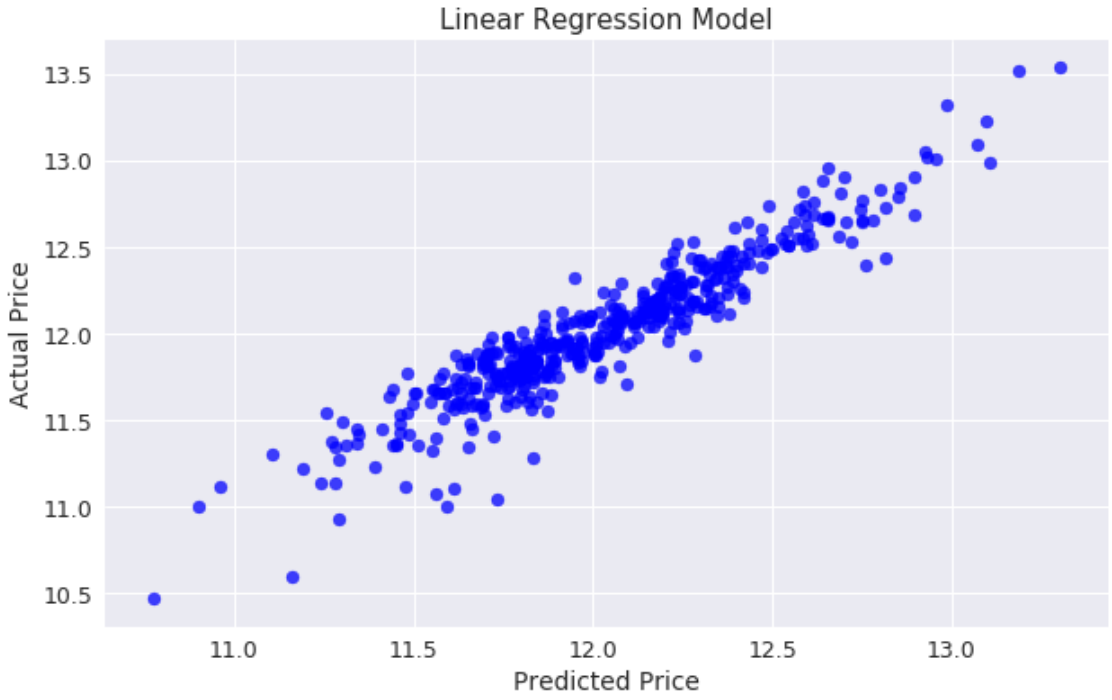
```
In [99]: predictions = model.predict(X_test)
```

```
In [100]: from sklearn.metrics import mean_squared_error
          print ('RMSE is: n', mean_squared_error(y_test, predictions))
```

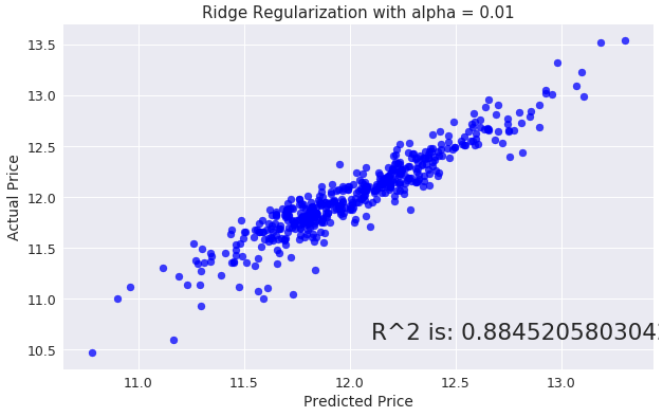
```
RMSE is: n 0.0196815767479
```

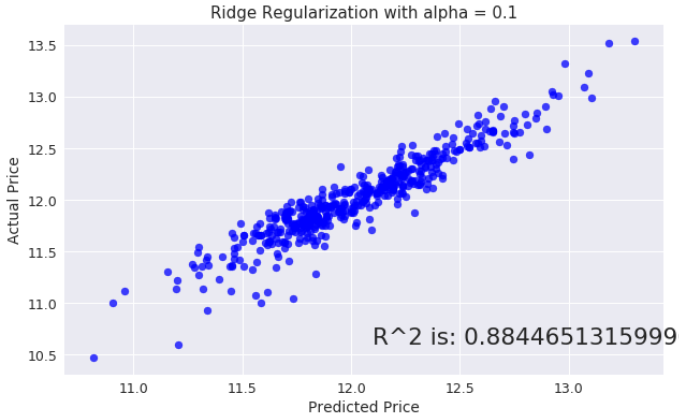
```
In [101]: actual_values = y_test
          plt.scatter(predictions, actual_values, alpha=.75,
```

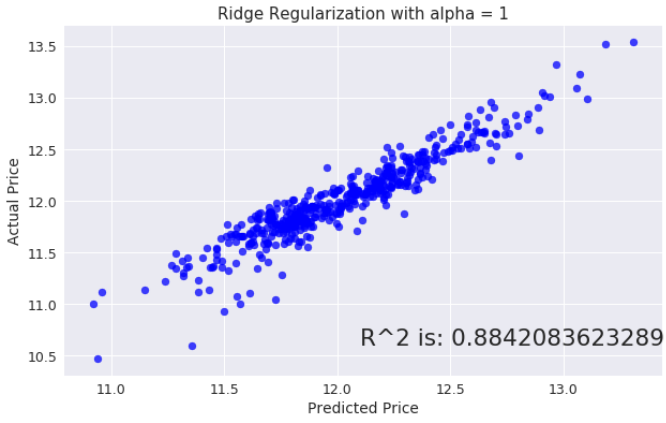
```
                color='b') #alpha helps to show overlapping data
plt.xlabel('Predicted Price')
plt.ylabel('Actual Price')
plt.title('Linear Regression Model')
plt.show()
```

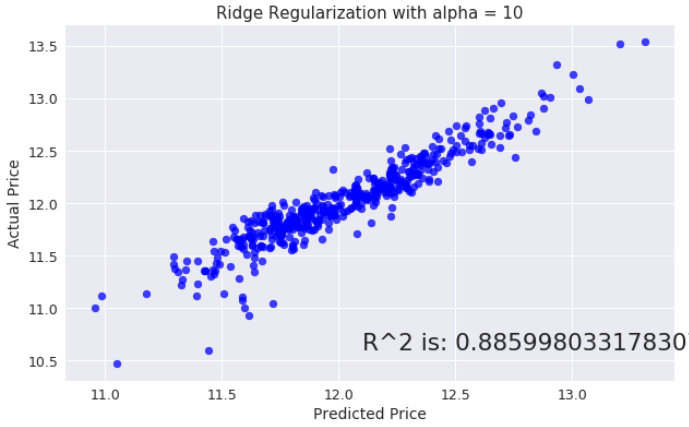


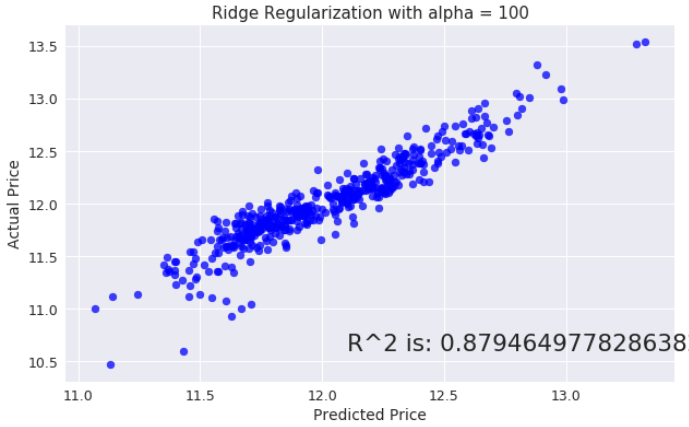

```
In [102]: for i in range (-2, 3):
              alpha = 10**i
              rm = linear_model.Ridge(alpha=alpha)
              ridge_model = rm.fit(X_train, y_train)
              preds_ridge = ridge_model.predict(X_test)

              plt.scatter(preds_ridge, actual_values, alpha=.75, color='b')
              plt.xlabel('Predicted Price')
              plt.ylabel('Actual Price')
              plt.title('Ridge Regularization with alpha = {}'.format(alpha))
              overlay = 'R^2 is: {}nRMSE is: {}'.format(
                            ridge_model.score(X_test, y_test),
                            mean_squared_error(y_test, preds_ridge))
              plt.annotate(s=overlay,xy=(12.1,10.6),size='x-large')
              plt.show()
```


Ridge Regularization with alpha = 0.01
Actual Price
Predicted Price
R^2 is: 0.8845205803042813nRMSE is: 0.019681317243139755


Ridge Regularization with alpha = 0.1
Actual Price
Predicted Price
R^2 is: 0.8844651315999665nRMSE is: 0.01969076744251915


Ridge Regularization with alpha = 1
Actual Price
Predicted Price
R^2 is: 0.8842083623289169nRMSE is: 0.019734528984576866


Ridge Regularization with alpha = 10
Actual Price
Predicted Price
R^2 is: 0.885998033178307nRMSE is: 0.019429512906036995


Ridge Regularization with alpha = 100
Actual Price
Predicted Price
R^2 is: 0.8794649778286382nRMSE is: 0.020542950566553673

In [ ]: