

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```
%matplotlib inline
import seaborn as sns
```

In [3]:

```
Data_Frame = pd.read_csv("PEP1.csv")
```

1. Understand the dataset:

a. Identify the shape of the dataset

In [4]:

```
Data_Frame.shape
```

Out[4]:

```
(1460, 81)
```

b. Identify variables with null values

In [5]:

```
Data_Null= Data_Frame.columns[Data_Frame.isnull().any()].to_list()  
Data_Null
```

Out[5]:

```
['LotFrontage',  
 'Alley',  
 'MasVnrType',  
 'MasVnrArea',  
 'BsmtQual',  
 'BsmtCond',  
 'BsmtExposure',  
 'BsmtFinType1',  
 'BsmtFinType2',  
 'Electrical',  
 'FireplaceQu',  
 'GarageType',  
 'GarageYrBlt',  
 'GarageFinish',  
 'GarageQual',  
 'GarageCond',  
 'PoolQC',  
 'Fence',  
 'MiscFeature']
```

c. Identify variables with unique values

In [6]:

```
for i in Data_Frame.columns:
    print (i , ":", Data_Frame[i].unique())
    print (" _ "*40)
    print (" _ "*40)
```

Id : [ 1 2 3 ... 1458 1459 1460]

MSSubClass : [ 60 20 70 50 190 45 90 120 30 85 80 160 75 180 40]

MSZoning : ['RL' 'RM' 'C (all)' 'FV' 'RH']

LotFrontage : [ 65. 80. 68. 60. 84. 85. 75. nan 51. 50. 70. 91. 7  
2. 66.  
101. 57. 44. 110. 98. 47. 108. 112. 74. 115. 61. 48. 33. 52.  
100. 24. 89. 63. 76. 81. 95. 69. 21. 32. 78. 121. 122. 40.  
105. 73. 77. 64. 94. 34. 90. 55. 88. 82. 71. 120. 107. 92.  
134. 62. 86. 141. 97. 54. 41. 79. 174. 99. 67. 83. 43. 103.  
93. 30. 129. 140. 35. 37. 118. 87. 116. 150. 111. 49. 96. 59.  
36. 56. 102. 58. 38. 109. 130. 53. 137. 45. 106. 104. 42. 39.  
144. 114. 128. 149. 313. 168. 182. 138. 160. 152. 124. 153. 46.]

LotArea : [ 8450 9600 11250 ... 17217 13175 9717]

Street : ['Pave' 'Grv1']

Alley : [nan 'Grv1' 'Pave']

LotShape : ['Reg' 'IR1' 'IR2' 'IR3']

LandContour : ['Lv1' 'Bnk' 'Low' 'HLS']

Utilities : ['AllPub' 'NoSeWa']

LotConfig : ['Inside' 'FR2' 'Corner' 'CulDSac' 'FR3']

LandSlope : ['Gtl' 'Mod' 'Sev']

Neighborhood : ['CollgCr' 'Veenker' 'Crawfor' 'NoRidge' 'Mitchel' 'Somerst'  
'NWAmes'  
'OldTown' 'BrkSide' 'Sawyer' 'NridgHt' 'mes' 'SawyerW' 'IDOTRR' 'MeadowV'  
'Edwards' 'Timber' 'Gilbert' 'StoneBr' 'ClearCr' 'NPkVill' 'Blmngtn'  
'BrDale' 'SWISU' 'Blueste']

Condition1 : ['Norm' 'Feedr' 'PosN' 'Artery' 'RRAe' 'RRNn' 'RRAn' 'PosA' 'RRN  
e']

Condition2 : ['Norm' 'Artery' 'RRNn' 'Feedr' 'PosN' 'PosA' 'RRAn' 'RRAe']

BldgType : ['1Fam' '2fmCon' 'Duplex' 'TwnhsE' 'Twnhs']

HouseStyle : ['2Story' '1Story' '1.5Fin' '1.5Unf' 'SFoyer' 'SLvl' '2.5Unf'  
'2.5Fin']

OverallQual : [ 7 6 8 5 9 4 10 3 1 2]

OverallCond : [5 8 6 7 4 2 3 9 1]

YearBuilt : [2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 1965 2005 1962  
2006  
1960 1929 1970 1967 1958 1930 2002 1968 2007 1951 1957 1927 1920 1966  
1959 1994 1954 1953 1955 1983 1975 1997 1934 1963 1981 1964 1999 1972  
1921 1945 1982 1998 1956 1948 1910 1995 1991 2009 1950 1961 1977 1985

1979	1885	1919	1990	1969	1935	1988	1971	1952	1936	1923	1924	1984	1926			
1940	1941	1987	1986	2008	1908	1892	1916	1932	1918	1912	1947	1925	1900			
1980	1989	1992	1949	1880	1928	1978	1922	1996	2010	1946	1913	1937	1942			
1938	1974	1893	1914	1906	1890	1898	1904	1882	1875	1911	1917	1872	1905]			
-	-	-	-	-	-	-	-	-	-	-	-	-	-			
-	-	-	-	-	-	-	-	-	-	-	-	-	-			
-	-	-	-	-	-	-	-	-	-	-	-	-	-			
YearRemodAdd	:	[	2003	1976	2002	1970	2000	1995	2005	1973	1950	1965	2006	1962	2007	1960
2001	1967	2004	2008	1997	1959	1990	1955	1983	1980	1966	1963	1987	1964			
1972	1996	1998	1989	1953	1956	1968	1981	1992	2009	1982	1961	1993	1999			
1985	1979	1977	1969	1958	1991	1971	1952	1975	2010	1984	1986	1994	1988			
1954	1957	1951	1978	1974]												
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RoofStyle	:	[	'Gable'	'Hip'	'Gambrel'	'Mansard'	'Flat'	'Shed']								
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
RoofMatl	:	[	'CompShg'	'WdShngl'	'Metal'	'WdShake'	'Membran'	'Tar&Grv'	'Roll'							
			'ClyTile']													
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Exterior1st	:	[	'VinylSd'	'MetalSd'	'Wd Sdng'	'HdBoard'	'BrkFace'	'WdShing'	'C							
emntBd'			'Plywood'	'AsbShng'	'Stucco'	'BrkComm'	'AsphShn'	'Stone'	'ImStucc'							
			'CBlock']													
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Exterior2nd	:	[	'VinylSd'	'MetalSd'	'Wd Shng'	'HdBoard'	'Plywood'	'Wd Sdng'	'C							
mentBd'			'BrkFace'	'Stucco'	'AsbShng'	'Brk Cmn'	'ImStucc'	'AsphShn'	'Stone'							
			'Other'	'CBlock']												
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MasVnrType	:	[	'BrkFace'	'None'	'Stone'	'BrkCmn'	nan]									
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MasVnrArea	:	[	1.960e+02	0.000e+00	1.620e+02	3.500e+02	1.860e+02	2.400e+02	2.8							
60e+02			3.060e+02	2.120e+02	1.800e+02	3.800e+02	2.810e+02	6.400e+02	2.000e+02							
			2.460e+02	1.320e+02	6.500e+02	1.010e+02	4.120e+02	2.720e+02	4.560e+02							
			1.031e+03	1.780e+02	5.730e+02	3.440e+02	2.870e+02	1.670e+02	1.115e+03							
			4.000e+01	1.040e+02	5.760e+02	4.430e+02	4.680e+02	6.600e+01	2.200e+01							
			2.840e+02	7.600e+01	2.030e+02	6.800e+01	1.830e+02	4.800e+01	2.800e+01							

3.360e+02 6.000e+02 7.680e+02 4.800e+02 2.200e+02 1.840e+02 1.129e+03  
1.160e+02 1.350e+02 2.660e+02 8.500e+01 3.090e+02 1.360e+02 2.880e+02  
7.000e+01 3.200e+02 5.000e+01 1.200e+02 4.360e+02 2.520e+02 8.400e+01  
6.640e+02 2.260e+02 3.000e+02 6.530e+02 1.120e+02 4.910e+02 2.680e+02  
7.480e+02 9.800e+01 2.750e+02 1.380e+02 2.050e+02 2.620e+02 1.280e+02  
2.600e+02 1.530e+02 6.400e+01 3.120e+02 1.600e+01 9.220e+02 1.420e+02  
2.900e+02 1.270e+02 5.060e+02 2.970e+02 nan 6.040e+02 2.540e+02  
3.600e+01 1.020e+02 4.720e+02 4.810e+02 1.080e+02 3.020e+02 1.720e+02  
3.990e+02 2.700e+02 4.600e+01 2.100e+02 1.740e+02 3.480e+02 3.150e+02  
2.990e+02 3.400e+02 1.660e+02 7.200e+01 3.100e+01 3.400e+01 2.380e+02  
1.600e+03 3.650e+02 5.600e+01 1.500e+02 2.780e+02 2.560e+02 2.250e+02  
3.700e+02 3.880e+02 1.750e+02 2.960e+02 1.460e+02 1.130e+02 1.760e+02  
6.160e+02 3.000e+01 1.060e+02 8.700e+02 3.620e+02 5.300e+02 5.000e+02  
5.100e+02 2.470e+02 3.050e+02 2.550e+02 1.250e+02 1.000e+02 4.320e+02  
1.260e+02 4.730e+02 7.400e+01 1.450e+02 2.320e+02 3.760e+02 4.200e+01  
1.610e+02 1.100e+02 1.800e+01 2.240e+02 2.480e+02 8.000e+01 3.040e+02  
2.150e+02 7.720e+02 4.350e+02 3.780e+02 5.620e+02 1.680e+02 8.900e+01  
2.850e+02 3.600e+02 9.400e+01 3.330e+02 9.210e+02 7.620e+02 5.940e+02  
2.190e+02 1.880e+02 4.790e+02 5.840e+02 1.820e+02 2.500e+02 2.920e+02  
2.450e+02 2.070e+02 8.200e+01 9.700e+01 3.350e+02 2.080e+02 4.200e+02  
1.700e+02 4.590e+02 2.800e+02 9.900e+01 1.920e+02 2.040e+02 2.330e+02  
1.560e+02 4.520e+02 5.130e+02 2.610e+02 1.640e+02 2.590e+02 2.090e+02  
2.630e+02 2.160e+02 3.510e+02 6.600e+02 3.810e+02 5.400e+01 5.280e+02  
2.580e+02 4.640e+02 5.700e+01 1.470e+02 1.170e+03 2.930e+02 6.300e+02  
4.660e+02 1.090e+02 4.100e+01 1.600e+02 2.890e+02 6.510e+02 1.690e+02  
9.500e+01 4.420e+02 2.020e+02 3.380e+02 8.940e+02 3.280e+02 6.730e+02  
6.030e+02 1.000e+00 3.750e+02 9.000e+01 3.800e+01 1.570e+02 1.100e+01  
1.400e+02 1.300e+02 1.480e+02 8.600e+02 4.240e+02 1.047e+03 2.430e+02  
8.160e+02 3.870e+02 2.230e+02 1.580e+02 1.370e+02 1.150e+02 1.890e+02  
2.740e+02 1.170e+02 6.000e+01 1.220e+02 9.200e+01 4.150e+02 7.600e+02  
2.700e+01 7.500e+01 3.610e+02 1.050e+02 3.420e+02 2.980e+02 5.410e+02  
2.360e+02 1.440e+02 4.230e+02 4.400e+01 1.510e+02 9.750e+02 4.500e+02  
2.300e+02 5.710e+02 2.400e+01 5.300e+01 2.060e+02 1.400e+01 3.240e+02  
2.950e+02 3.960e+02 6.700e+01 1.540e+02 4.250e+02 4.500e+01 1.378e+03  
3.370e+02 1.490e+02 1.430e+02 5.100e+01 1.710e+02 2.340e+02 6.300e+01  
7.660e+02 3.200e+01 8.100e+01 1.630e+02 5.540e+02 2.180e+02 6.320e+02  
1.140e+02 5.670e+02 3.590e+02 4.510e+02 6.210e+02 7.880e+02 8.600e+01  
7.960e+02 3.910e+02 2.280e+02 8.800e+01 1.650e+02 4.280e+02 4.100e+02  
5.640e+02 3.680e+02 3.180e+02 5.790e+02 6.500e+01 7.050e+02 4.080e+02  
2.440e+02 1.230e+02 3.660e+02 7.310e+02 4.480e+02 2.940e+02 3.100e+02  
2.370e+02 4.260e+02 9.600e+01 4.380e+02 1.940e+02 1.190e+02]

- - - - -  
- - - - -  
- - - - -

ExterQual : ['Gd' 'TA' 'Ex' 'Fa']

- - - - -  
- - - - -  
- - - - -

ExterCond : ['TA' 'Gd' 'Fa' 'Po' 'Ex']

- - - - -  
- - - - -  
- - - - -

Foundation : ['PConc' 'CBlock' 'BrkTil' 'Wood' 'Slab' 'Stone']

- - - - -

```

- - - - -
- - - - -
BsmtQual : ['Gd' 'TA' 'Ex' nan 'Fa']
- - - - -
- - - - -
BsmtCond : ['TA' 'Gd' nan 'Fa' 'Po']
- - - - -
- - - - -
BsmtExposure : ['No' 'Gd' 'Mn' 'Av' nan]
- - - - -
- - - - -
BsmtFinType1 : ['GLQ' 'ALQ' 'Unf' 'Rec' 'BLQ' nan 'LwQ']
- - - - -
- - - - -
BsmtFinSF1 : [ 706  978  486  216  655  732 1369  859    0  851  906  998  73
7  733
  578  646  504  840  188  234 1218 1277 1018 1153 1213  731  643  967
  747  280  179  456 1351   24  763  182  104 1810  384  490  649  632
  941  739  912 1013  603 1880  565  320  462  228  336  448 1201   33
  588  600  713 1046  648  310 1162  520  108  569 1200  224  705  444
  250  984   35  774  419  170 1470  938  570  300  120  116  512  567
  445  695  405 1005  668  821  432 1300  507  679 1332  209  680  716
1400  416  429  222   57  660 1016  370  351  379 1288  360  639  495
  288 1398  477  831 1904  436  352  611 1086  297  626  560  390  566
1126 1036 1088  641  617  662  312 1065  787  468   36  822  378  946
  341   16  550  524   56  321  842  689  625  358  402   94 1078  329
  929  697 1573  270  922  503 1334  361  672  506  714  403  751  226
  620  546  392  421  905  904  430  614  450  210  292  795 1285  819
  420  841  281  894 1464  700  262 1274  518 1236  425  692  987  970
   28  256 1619   40  846 1124  720  828 1249  810  213  585  129  498
1270  573 1410 1082  236  388  334  874  956  773  399  162  712  609
  371  540   72  623  428  350  298 1445  218  985  631 1280  241  690
  266  777  812  786 1116  789 1056   50 1128  775 1309 1246  986  616
1518  664  387  471  385  365 1767  133  642  247  331  742 1606  916
  185  544  553  326  778  386  426  368  459 1350 1196  630  994  168
1261 1567  299  897  607  836  515  374 1231  111  356  400  698 1247
  257  380   27  141  991  650  521 1436 2260  719  377 1330  348 1219
  783  969  673 1358 1260  144  584  554 1002  619  180  559  308  866
  895  637  604 1302 1071  290  728   2 1441  943  231  414  349  442
  328  594  816 1460 1324 1338  685 1422 1283   81  454  903  605  990
  206  150  457   48  871   41  674  624  480 1154  738  493 1121  282
  500  131 1696  806 1361  920 1721  187 1138  988  193  551  767 1186
  892  311  827  543 1003 1059  239  945   20 1455  965  980  863  533
1084 1173  523 1148  191 1234  375  808  724  152 1180  252  832  575
  919  439  381  438  549  612 1163  437  394 1416  422  762  975 1097
  251  686  656  568  539  862  197  516  663  608 1636  784  249 1040
  483  196  572  338  330  156 1390  513  460  659  364  564  306  505
  932  750   64  633 1170  899  902 1238  528 1024 1064  285 2188  465

```



```
322 860 599 354 63 223 301 443 489 284 294 814 165 552
833 464 936 772 1440 748 982 398 562 484 417 699 696 896
556 1106 651 867 854 1646 1074 536 1172 915 595 1237 273 684
324 1165 138 1513 317 1012 1022 509 900 1085 1104 240 383 644
397 740 837 220 586 535 410 75 824 592 1039 510 423 661
248 704 412 1032 219 708 415 1004 353 702 369 622 212 645
852 1150 1258 275 176 296 538 1157 492 1198 1387 522 658 1216
1480 2096 1159 440 1456 883 547 788 485 340 1220 427 344 756
1540 666 803 1000 885 1386 319 534 125 1314 602 192 593 804
1053 532 1158 1014 194 167 776 5644 694 1572 746 1406 925 482
189 765 80 1443 259 735 734 1447 548 315 1282 408 309 203
865 204 790 1320 769 1070 264 759 1373 976 781 25 1110 404
580 678 958 1336 1079 49 830]
```

```
- - - - -
- - - - -
- - - - -
BsmtFinType2 : ['Unf' 'BLQ' nan 'ALQ' 'Rec' 'LwQ' 'GLQ']
- - - - -
- - - - -
- - - - -
```

```
BsmtFinSF2 : [ 0 32 668 486 93 491 506 712 362 41 169 869 15
0 670
28 1080 181 768 215 374 208 441 184 279 306 180 580 690
692 228 125 1063 620 175 820 1474 264 479 147 232 380 544
294 258 121 391 531 344 539 713 210 311 1120 165 532 96
495 174 1127 139 202 645 123 551 219 606 612 480 182 132
336 468 287 35 499 723 119 40 117 239 80 472 64 1057
127 630 128 377 764 345 1085 435 823 500 290 324 634 411
841 1061 466 396 354 149 193 273 465 400 682 557 230 106
791 240 547 469 177 108 600 492 211 168 1031 438 375 144
81 906 608 276 661 68 173 972 105 420 546 334 352 872
110 627 163 1029]
```

```
- - - - -
- - - - -
- - - - -
BsmtUnfSF : [ 150 284 434 540 490 64 317 216 952 140 134 177 175
1494
```

```
520 832 426 0 468 525 1158 637 1777 200 204 1566 180 486
207 649 1228 1234 380 408 1117 1097 84 326 445 383 167 465
1296 83 1632 736 192 612 816 32 935 321 860 1410 148 217
530 1346 576 318 1143 1035 440 747 701 343 280 404 840 724
295 1768 448 36 1530 1065 384 1288 684 1013 402 635 163 168
176 370 350 381 410 741 1226 1053 641 516 793 1139 550 905
104 310 252 1125 203 728 732 510 899 1362 30 958 556 413
479 297 658 262 891 1304 519 1907 336 107 432 403 811 396
970 506 884 400 896 253 409 93 1200 572 774 769 1335 340
882 779 112 470 294 1686 360 441 354 700 725 320 554 312
968 504 1107 577 660 99 871 474 289 600 755 625 1121 276
186 1424 1140 375 92 305 1176 78 274 311 710 686 457 1232
1498 1010 160 2336 630 638 162 70 1357 1194 773 483 235 125
1390 594 1694 488 357 626 916 1020 1367 798 452 392 975 361
270 602 1482 680 606 88 342 212 1095 96 628 1560 744 2121
768 386 1468 1145 244 698 1079 570 476 131 184 143 1092 324
1541 1470 536 319 599 622 179 292 286 80 712 291 153 1088
```



1845 1020 1367 1444 1573 1302 1314 975 1604 963 1482 506 926 1422  
802 740 1095 1385 1152 1240 1560 2121 1160 807 1468 1575 625 858  
698 1079 768 795 1416 1003 702 1165 1470 2000 700 319 861 1896  
697 972 2136 716 1347 1372 1249 1136 1502 1162 710 1719 1383 844  
596 1056 3206 1358 943 1499 1922 1536 1208 1215 967 721 1684 536  
958 1478 764 1848 1869 616 624 940 1142 1062 888 883 1394 1099  
1268 953 744 608 847 683 870 1580 1856 982 1026 1293 939 784  
1256 658 1041 1682 804 788 1144 961 1260 1310 1141 806 1281 1034  
1276 1340 1344 988 651 1518 907 901 765 799 648 3094 1440 1258  
915 1517 930 813 1533 872 1242 1364 588 709 560 1375 1277 1626  
1488 808 547 1976 2153 1705 1833 1792 1216 999 1113 1073 954 264  
1269 190 3200 866 1501 777 1218 1368 1084 2006 1244 3138 1379 1257  
1452 528 2035 611 707 880 1051 1581 1838 1650 723 654 1204 1069  
1709 998 993 1374 1389 1163 1122 1496 846 372 1164 1050 2042 1868  
1437 742 770 1722 1814 1430 1058 908 600 965 1032 1299 1120 936  
783 1822 1522 980 1116 978 1156 636 1554 1386 811 1520 1952 1766  
981 1094 2109 525 776 1486 1629 1138 2077 1406 1021 1408 738 1477  
2046 923 1291 1195 1190 874 551 1419 2444 1210 927 1112 1391 1800  
360 1473 1643 1324 270 859 718 1176 1311 971 1742 941 1698 1584  
1595 868 1153 893 1349 1337 1720 1479 1030 1318 1252 983 1860 836  
1935 1614 761 1413 956 712 650 773 1926 731 1417 1024 849 1442  
1649 1568 778 1489 2078 1454 1516 1067 1559 1127 1390 1273 918 1763  
1090 1054 1039 1148 1002 1638 105 676 1184 1109 892 2217 1505 1059  
951 2330 1670 1623 1017 1105 1001 546 480 1134 1104 1272 1316 1126  
1181 1753 964 1466 925 1905 1500 585 1632 819 1616 1161 828 945  
979 561 696 1330 817 1098 1428 673 1241 944 1225 1266 1128 485  
1930 1396 916 822 750 1700 1007 1187 691 1574 1680 1346 985 1657  
602 1022 1082 810 1504 1220 1132 1565 1338 1654 1620 1055 800 1306  
1475 2524 1992 1193 973 854 662 1103 1154 942 1048 727 690 1096  
1459 1251 1247 1074 1271 290 655 1463 1836 803 833 408 533 1012  
1552 1005 1530 974 1567 1006 1042 1298 704 932 1219 1296 1198 959  
1261 1598 1683 818 1600 2396 1624 831 1224 663 879 815 1630 2158  
931 1660 559 1300 1702 1075 1361 1106 1476 1689 2076 792 2110 1405  
1192 746 1986 841 2002 1332 935 1019 661 1309 1328 1085 6110 1246  
771 976 1652 1278 1902 1274 1393 1622 1352 420 1795 544 1510 911  
693 1284 1732 2033 570 1980 814 873 757 1108 2633 1571 984 1205  
714 1746 1525 482 1356 862 839 1286 1485 1594 622 791 708 1223  
913 656 1319 1932 539 1221 1542]

- - - - -  
- - - - -  
- - - - -

Heating : ['GasA' 'GasW' 'Grav' 'Wall' 'OthW' 'Floor']

- - - - -  
- - - - -  
- - - - -

HeatingQC : ['Ex' 'Gd' 'TA' 'Fa' 'Po']

- - - - -  
- - - - -  
- - - - -

CentralAir : ['Y' 'N']

- - - - -  
- - - - -  
- - - - -  
- - - - -

Electrical : ['SBrkr' 'FuseF' 'FuseA' 'FuseP' 'Mix' nan]

1stFlrSF : [ 856 1262 920 961 1145 796 1694 1107 1022 1077 1040 1182 912  
1494

1253	854	1004	1296	1114	1339	1158	1108	1795	1060	1600	900	1704	520
649	1228	1234	1700	1561	1132	1097	1297	1057	1152	1324	1328	884	938
1150	1752	1518	1656	736	955	794	816	1842	1360	1425	983	860	1426
780	581	1370	902	1143	2207	1479	747	1304	2223	845	885	1086	840
526	952	1072	1768	682	1337	1563	1065	804	1301	684	612	1013	990
1235	964	1260	905	680	1588	960	835	1225	1610	977	1535	1226	1053
1047	789	997	1844	1216	774	1282	2259	1436	729	1092	1125	1699	728
988	772	1080	1199	1586	958	660	1327	1721	1682	1214	1959	928	864
1734	910	1501	1728	970	875	896	969	1710	1252	1200	572	991	1392
1232	1572	1541	882	1149	808	1867	1707	1064	1362	1651	2158	1164	2234
968	769	901	1340	936	1217	1224	1593	1549	725	1431	855	1726	929
1713	1121	1279	865	848	720	1442	1696	1100	1180	1212	932	689	1236
810	1137	1248	1498	1010	811	2392	630	483	1555	1194	1490	894	1414
1014	798	1566	866	889	626	1222	1872	908	1375	1444	1306	1625	1302
1314	1005	1604	963	1382	1482	926	764	1422	802	1052	778	1113	1095
1363	1632	1560	2121	1156	1175	1468	1575	625	1085	858	698	1079	1148
1644	1003	975	1041	1336	1210	1675	2000	1122	1035	861	1944	697	972
793	2036	832	716	1153	1088	1372	1472	1249	1136	1553	1163	1898	803
1719	1383	1445	596	1056	1629	1358	943	1619	1922	1536	1621	1215	993
841	1684	536	1478	1848	1869	1453	616	1192	1167	1142	1352	495	790
672	1394	1268	1287	953	1120	752	1319	847	904	914	1580	1856	1007
1026	939	784	1269	658	1742	788	735	1144	876	1112	1288	1310	1165
806	1620	1166	1071	1050	1276	1028	756	1344	1602	1470	1196	707	907
1208	1412	765	827	734	694	2402	1440	1128	1258	933	1689	1888	956
679	813	1533	888	786	1242	624	1663	833	979	575	849	1277	1634
1502	1161	1976	1652	1493	2069	1718	1131	1850	1792	916	999	1073	1484
1766	886	3228	1133	899	1801	1218	1368	2020	1378	1244	3138	1266	1476
605	2515	1509	751	334	820	880	1159	1601	1838	1680	767	664	1377
915	768	825	1069	1717	1126	1006	1048	897	1557	1389	996	1134	1496
846	576	877	1320	703	1429	2042	1521	989	2028	838	1473	779	770
924	1826	1402	1647	1058	927	600	1186	1940	1029	1032	1299	1054	807
1828	1548	980	1012	1116	1520	1350	1089	1554	1411	800	1567	981	1094
1051	822	755	909	2113	525	851	1486	1686	1181	2097	1454	1465	1679
1437	738	1839	792	2046	923	1291	1668	1195	1190	874	551	1419	2444
1238	1067	1391	1800	1264	372	1824	859	1576	1178	1325	971	1698	1776
1616	1146	948	1349	1464	1720	1038	742	757	1506	1836	1690	1220	1117
1973	1204	1614	1430	1110	1342	966	976	1062	1127	1285	773	1966	1428
1075	1309	1044	686	1661	1008	944	1489	2084	1434	1160	941	1516	1559
1099	1701	1307	1456	918	1779	702	1512	1039	1002	1646	1547	1036	676
1184	1462	1155	1090	1187	954	892	1709	1712	872	2217	1505	1068	951
2364	1670	1063	1636	1020	1105	1015	1001	546	480	1229	1272	1316	1617
1098	1788	1466	925	1905	1500	1207	1188	1381	965	1168	561	696	1542
824	783	673	869	1241	1118	1407	750	691	1574	1504	985	1657	1664
1082	2898	1687	1654	1055	1803	1532	2524	1733	1992	1771	930	1526	1091
1523	1364	1130	1096	1338	1103	1154	799	893	829	1240	1459	1251	1247
1390	438	950	887	1021	1552	812	1530	974	986	1042	1298	1811	1265
1640	1432	959	1831	1261	1170	2129	818	1124	2411	949	1624	831	1622
842	663	879	815	1630	1074	2196	1283	1660	1318	1211	2136	1138	1702
1507	1361	1024	1141	1173	2076	1140	1034	2110	1405	760	1987	1104	713
2018	1968	1332	935	1357	661	1724	1573	1582	1659	4692	1246	753	1203

1294 1902 1274 1787 1061 708 1584 1334 693 1284 1172 2156 2053 992  
1078 1980 1281 814 2633 1571 984 754 2117 998 1416 1746 1525 1221  
741 1569 1223 962 1537 1932 1423 913 1578 2073 1256]

- - - - -  
- - - - -  
- - - - -  
2ndFlrSF : [ 854 0 866 756 1053 566 983 752 1142 1218 668 1320 631  
716

676 860 1519 530 808 977 1330 833 765 462 213 548 960 670  
1116 876 612 1031 881 790 755 592 939 520 639 656 1414 884  
729 1523 728 351 688 941 1032 848 836 475 739 1151 448 896  
524 1194 956 1070 1096 467 547 551 880 703 901 720 316 1518  
704 1178 754 601 1360 929 445 564 882 920 518 817 1257 741  
672 1306 504 1304 1100 730 689 591 888 1020 828 700 842 1286  
864 829 1092 709 844 1106 596 807 625 649 698 840 780 568  
795 648 975 702 1242 1818 1121 371 804 325 809 1200 871 1274  
1347 1332 1177 1080 695 167 915 576 605 862 495 403 838 517  
1427 784 711 468 1081 886 793 665 858 874 526 590 406 1157  
299 936 438 1098 766 1101 1028 1017 1254 378 1160 682 110 600  
678 834 384 512 930 868 224 1103 560 811 878 574 910 620  
687 546 902 1000 846 1067 914 660 1538 1015 1237 611 707 527  
1288 832 806 1182 1040 439 717 511 1129 1370 636 533 745 584  
812 684 595 988 800 677 573 1066 778 661 1440 872 788 843  
713 567 651 762 482 738 586 679 644 900 887 1872 1281 472  
1312 319 978 1093 473 664 1540 1276 441 348 1060 714 744 1203  
783 1097 734 767 1589 742 686 1128 1111 1174 787 1072 1088 1063  
545 966 623 432 581 540 769 1051 761 779 514 455 1426 785  
521 252 813 1120 1037 1169 1001 1215 928 1140 1243 571 1196 1038  
561 979 701 332 368 883 1336 1141 634 912 798 985 826 831  
750 456 602 855 336 408 980 998 1168 1208 797 850 898 1054  
895 954 772 1230 727 454 370 628 304 582 1122 1134 885 640  
580 1112 653 220 240 1362 534 539 650 918 933 712 1796 971  
1175 743 523 1216 2065 272 685 776 630 984 875 913 464 1039  
1259 940 892 725 924 764 925 1479 192 589 992 903 430 748  
587 994 950 1323 732 1357 557 1296 390 1185 873 1611 457 796  
908 550 989 932 358 1392 349 691 1349 768 208 622 857 556  
1044 708 626 904 510 1104 830 981 870 694 1152]

- - - - -  
- - - - -  
- - - - -  
LowQualFinSF : [ 0 360 513 234 528 572 144 392 371 390 420 473 156 515 80  
53 232 481  
120 514 397 479 205 384]

- - - - -  
- - - - -  
- - - - -  
GrLivArea : [1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 1040 2324 912  
1494

1253 854 1004 1296 1114 1339 2376 1108 1795 1060 1600 900 1704 520  
1317 1228 1234 1700 1561 2452 1097 1297 1057 1152 1324 1328 884 938  
1150 1752 2149 1656 1452 955 1470 1176 816 1842 1360 1425 1739 1720  
2945 780 1158 1111 1370 2034 2473 2207 1479 747 2287 2223 845 1718  
1086 1605 988 952 1285 1768 1230 2142 1337 1563 1065 1474 2417 1560  
1224 1526 990 1235 964 2291 1588 960 835 1225 1610 1732 1535 1226

1818 1992 1047 789 1517 1844 1855 1430 2696 2259 2320 1458 1092 1125  
3222 1456 1123 1080 1199 1586 754 958 840 1348 1053 2157 2054 1327  
1721 1682 1214 1959 1852 1764 864 1734 1385 1501 1728 1709 875 2035  
1344 969 1993 1252 1200 1096 1968 1947 2462 1232 2668 1541 882 1616  
1355 1867 2161 1707 1382 1767 1651 2158 2060 1920 2234 968 1525 1802  
1340 2082 3608 1217 1593 2727 1431 1726 3112 2229 1713 1121 1279 1310  
848 1284 1442 1696 1100 2062 1212 1392 1236 1436 1954 1248 1498 2267  
1552 2392 1302 2520 987 1555 1194 2794 894 1960 1414 1744 1487 1566  
866 1440 2110 1872 1928 1375 1668 2144 1306 1625 1640 1314 1604 1792  
2574 1316 764 1422 1511 2192 778 1113 1939 1363 2270 1632 1548 2121  
2022 1982 1468 1575 1250 858 1396 1919 1716 2263 1644 1003 1558 1950  
1743 1336 3493 2000 2243 1406 861 1944 972 1118 2036 1641 1432 2353  
2646 1472 2596 2468 2730 1163 2978 803 1719 1383 2134 1192 1056 1629  
1358 1638 1922 1536 1621 1215 1908 841 1684 1112 1577 1478 1626 2728  
1869 1453 720 1595 1167 1142 1352 1924 1505 1574 1394 1268 1287 1664  
752 1319 904 914 2466 1856 1800 1691 1301 1797 784 1953 1269 1184  
2332 1367 1961 788 1034 1144 1812 1550 1288 672 1572 1620 1639 1680  
2172 2078 1276 1028 2097 1400 2624 1134 1602 2630 1196 1389 907 1208  
1412 1198 1365 630 1661 694 2402 1573 1258 1689 1888 1886 1376 1183  
813 1533 1756 1590 1242 1663 1666 1203 1935 1135 1660 1277 1634 1502  
1969 1072 1976 1652 970 1493 2643 1131 1850 1826 1216 999 1073 1484  
2414 1304 1578 886 3228 1820 899 1218 1801 1322 1911 1378 1041 1368  
2020 2119 2344 1796 2080 1294 1244 4676 2398 1266 928 2713 605 2515  
1509 827 334 1347 1724 1159 1601 1838 2285 767 1496 2183 1635 768  
825 2094 1069 1126 2046 1048 1446 1557 996 1674 2295 1647 2504 2132  
943 1692 1109 1477 1320 1429 2042 2775 2028 838 860 1473 935 1582  
2296 924 1402 1556 1904 1915 1986 2008 3194 1029 2153 1032 1120 1054  
832 1828 2262 2614 980 1512 1790 1116 1520 1350 1750 1554 1411 3395  
800 1387 796 1567 1518 1929 2704 1766 981 1094 1839 1665 1510 1469  
2113 1486 2448 1181 1936 2380 1679 1437 1180 1476 1369 1136 1441 792  
923 1291 1761 1102 1419 4316 2519 1539 1137 616 1148 1391 1164 2576  
1824 729 1178 2554 2418 971 1742 1698 1776 1146 2031 948 1349 1464  
2715 2256 2640 1529 1140 2098 1026 1471 1386 2531 1547 2365 1506 1714  
1836 3279 1220 1117 1973 1204 1614 1603 1110 1342 2084 901 2087 1145  
1062 2013 1895 1564 773 3140 1688 2822 1128 1428 1576 2138 1309 1044  
1008 1052 936 1733 1489 1434 2126 1223 1829 1516 1067 1559 1099 1482  
1165 1416 1701 1775 2358 1646 1445 1779 1481 2654 1426 1039 1372 1002  
1949 910 2610 2224 1155 1090 2230 892 1712 1393 2217 1683 1068 951  
2240 2364 1670 902 1063 1636 2057 2274 1015 2002 480 1229 2127 2200  
1617 1686 2374 1978 1788 2236 1466 925 1905 1500 2069 1971 1962 2403  
1381 965 1958 2872 1894 1308 1098 1095 918 2019 869 1241 2612 2290  
1940 2030 1851 1050 944 691 1504 985 1657 1522 1271 1022 1082 1132  
2898 1264 3082 1654 954 1803 2329 2524 2868 1771 930 1977 1989 1523  
1364 2184 1991 1338 2337 1103 1154 2260 1571 1611 2521 893 1240 1740  
1459 1251 1247 1088 438 950 2622 2021 1690 1658 1964 833 1012 698  
1005 1530 1981 974 2210 986 1020 1868 2828 1006 1298 932 1811 1265  
1580 1876 1671 2108 3627 1261 3086 2345 1343 1124 2514 4476 1130 1221  
1699 1624 1804 1622 1863 1630 1074 2196 1283 1845 1902 1211 1846 2136  
1490 1138 1933 1702 1507 2620 1190 1188 1784 1948 1141 1173 2076 1553  
2058 1405 874 2167 1987 1166 1675 1889 2018 3447 1524 1357 1395 2447  
1659 1970 2372 5642 1246 1983 2526 1708 1122 1274 2810 2599 2112 1787  
1923 708 774 2792 1334 693 1861 872 2169 1913 2156 2634 3238 1865  
1078 1980 2601 1738 1475 1374 2633 790 2117 1762 2784 1746 1584 1912  
2482 1687 1513 1608 2093 1840 1848 1569 2450 2201 804 1537 1932 1725  
2555 2007 913 1346 2073 2340 1256]

- - - - -  
- - - - -

BsmtFullBath : [1 0 2 3]

BsmtHalfBath : [0 1 2]

FullBath : [2 1 3 0]

HalfBath : [1 0 2]

BedroomAbvGr : [3 4 1 2 0 5 6 8]

KitchenAbvGr : [1 2 3 0]

KitchenQual : ['Gd' 'TA' 'Ex' 'Fa']

TotRmsAbvGrd : [ 8 6 7 9 5 11 4 10 12 3 2 14]

Function1 : ['Typ' 'Min1' 'Maj1' 'Min2' 'Mod' 'Maj2' 'Sev']

Fireplaces : [0 1 2 3]

FireplaceQu : [nan 'TA' 'Gd' 'Fa' 'Ex' 'Po']

GarageType : ['Attchd' 'Detchd' 'BuiltIn' 'CarPort' nan 'Basment' '2Types']

GarageYrBlt : [2003. 1976. 2001. 1998. 2000. 1993. 2004. 1973. 1931. 1939. 1965. 2005.

1962. 2006. 1960. 1991. 1970. 1967. 1958. 1930. 2002. 1968. 2007. 2008.  
1957. 1920. 1966. 1959. 1995. 1954. 1953. nan 1983. 1977. 1997. 1985.  
1963. 1981. 1964. 1999. 1935. 1990. 1945. 1987. 1989. 1915. 1956. 1948.  
1974. 2009. 1950. 1961. 1921. 1900. 1979. 1951. 1969. 1936. 1975. 1971.  
1923. 1984. 1926. 1955. 1986. 1988. 1916. 1932. 1972. 1918. 1980. 1924.  
1996. 1940. 1949. 1994. 1910. 1978. 1982. 1992. 1925. 1941. 2010. 1927.  
1947. 1937. 1942. 1938. 1952. 1928. 1922. 1934. 1906. 1914. 1946. 1908.  
1929. 1933.]

GarageFinish : ['RFn' 'Unf' 'Fin' nan]

GarageCars : [2 3 1 0 4]

GarageArea : [ 548 460 608 642 836 480 636 484 468 205 384 736 352 840

576 516 294 853 280 534 572 270 890 772 319 240 250 271  
447 556 691 672 498 246 0 440 308 504 300 670 826 386  
388 528 894 565 641 288 645 852 558 220 667 360 427 490  
379 297 283 509 405 758 461 400 462 420 432 506 684 472  
366 476 410 740 648 273 546 325 792 450 180 430 594 390  
540 264 530 435 453 750 487 624 471 318 766 660 470 720  
577 380 434 866 495 564 312 625 680 678 726 532 216 303  
789 511 616 521 451 1166 252 497 682 666 786 795 856 473  
398 500 349 454 644 299 210 431 438 675 968 721 336 810  
494 457 818 463 604 389 538 520 309 429 673 884 868 492  
413 924 1053 439 671 338 573 732 505 575 626 898 529 685  
281 539 418 588 282 375 683 843 552 870 888 746 708 513  
1025 656 872 292 441 189 880 676 301 474 706 617 445 200  
592 566 514 296 244 610 834 639 501 846 560 596 600 373  
947 350 396 864 304 784 696 569 628 550 493 578 198 422  
228 526 525 908 499 508 694 874 164 402 515 286 603 900  
583 889 858 502 392 403 527 765 367 426 615 871 570 406  
590 612 650 1390 275 452 842 816 621 544 486 230 261 531  
393 774 749 364 627 260 256 478 442 562 512 839 330 711  
1134 416 779 702 567 832 326 551 606 739 408 475 704 983  
768 632 541 320 800 831 554 878 752 614 481 496 423 841  
895 412 865 630 605 602 618 444 397 455 409 820 1020 598  
857 595 433 776 1220 458 613 456 436 812 686 611 425 343  
479 619 902 574 523 414 738 354 483 327 756 690 284 833  
601 533 522 788 555 689 796 808 510 255 424 305 368 824  
328 160 437 665 290 912 905 542 716 586 467 582 1248 1043



[illegible]

```

141 15 126 236]
- - - - -
- - - - -
- - - - -
EnclosedPorch : [ 0 272 228 205 176 87 172 102 37 144 64 114 202 128 156
44 77 192
140 180 183 39 184 40 552 30 126 96 60 150 120 112 252 52 224 234
244 268 137 24 108 294 177 218 242 91 160 130 169 105 34 248 236 32
80 115 291 116 158 210 36 200 84 148 136 240 54 100 189 293 164 216
239 67 90 56 129 98 143 70 386 154 185 134 196 264 275 230 254 68
194 318 48 94 138 226 174 19 170 220 214 280 190 330 208 145 259 81
42 123 162 286 168 20 301 198 221 212 50 99]
- - - - -
- - - - -
- - - - -
3SsnPorch : [ 0 320 407 130 180 168 140 508 238 245 196 144 182 162 23 216
96 153
290 304]
- - - - -
- - - - -
- - - - -
ScreenPorch : [ 0 176 198 291 252 99 184 168 130 142 192 410 224 266 170 15
4 153 144
128 259 160 271 234 374 185 182 90 396 140 276 180 161 145 200 122 95
120 60 126 189 260 147 385 287 156 100 216 210 197 204 225 152 175 312
222 265 322 190 233 63 53 143 273 288 263 80 163 116 480 178 440 155
220 119 165 40]
- - - - -
- - - - -
- - - - -
PoolArea : [ 0 512 648 576 555 480 519 738]
- - - - -
- - - - -
- - - - -
PoolQC : [nan 'Ex' 'Fa' 'Gd']
- - - - -
- - - - -
- - - - -
Fence : [nan 'MnPrv' 'GdWo' 'GdPrv' 'MnWw']
- - - - -
- - - - -
- - - - -
MiscFeature : [nan 'Shed' 'Gar2' 'Othr' 'TenC']
- - - - -
- - - - -
- - - - -
MiscVal : [ 0 700 350 500 400 480 450 15500 1200 800 2000
600
3500 1300 54 620 560 1400 8300 1150 2500]

```

```

- - - - -
- - - - -
- - - - -
MoSold : [ 2 5 9 12 10 8 11 4 1 7 3 6]
- - - - -
- - - - -
YrSold : [2008 2007 2006 2009 2010]
- - - - -
- - - - -
SaleType : ['WD' 'New' 'COD' 'ConLD' 'ConLI' 'CWD' 'ConLw' 'Con' 'Oth']
- - - - -
- - - - -
SaleCondition : ['Normal' 'Abnorml' 'Partial' 'AdjLand' 'Alloca' 'Family']
- - - - -
- - - - -
SalePrice : [208500 181500 223500 140000 250000 143000 307000 200000 129900 1
18000
129500 345000 144000 279500 157000 132000 149000 90000 159000 139000
325300 139400 230000 154000 256300 134800 306000 207500 68500 40000
149350 179900 165500 277500 309000 145000 153000 109000 82000 160000
170000 130250 141000 319900 239686 249700 113000 127000 177000 114500
110000 385000 130000 180500 172500 196500 438780 124900 158000 101000
202500 219500 317000 180000 226000 80000 225000 244000 185000 144900
107400 91000 135750 136500 193500 153500 245000 126500 168500 260000
174000 164500 85000 123600 109900 98600 163500 133900 204750 214000
94750 83000 128950 205000 178000 118964 198900 169500 100000 115000
190000 136900 383970 217000 259500 176000 155000 320000 163990 136000
153900 181000 84500 128000 87000 150000 150750 220000 171000 231500
166000 204000 125000 105000 222500 122000 372402 235000 79000 109500
269500 254900 162500 412500 103200 152000 127500 325624 183500 228000
128500 215000 239000 163000 184000 243000 211000 501837 200100 120000
475000 173000 135000 153337 286000 315000 192000 148500 311872 104000
274900 171500 112000 143900 277000 98000 186000 252678 156000 161750
134450 210000 107000 311500 167240 204900 97000 386250 290000 106000
192500 148000 403000 94500 128200 216500 89500 185500 194500 318000
262500 110500 241500 137000 76500 276000 151000 73000 175500 179500
120500 266000 124500 201000 415298 228500 244600 179200 164700 88000
153575 233230 135900 131000 167000 142500 175000 158500 267000 149900
295000 305900 82500 360000 165600 119900 375000 188500 270000 187500
342643 354000 301000 126175 242000 324000 145250 214500 78000 119000
284000 207000 228950 377426 202900 87500 140200 151500 157500 437154
318061 95000 105900 177500 134000 280000 198500 147000 165000 162000
172400 134432 123000 61000 340000 394432 179000 187750 213500 76000
240000 81000 191000 426000 106500 129000 67000 241000 245500 164990
108000 258000 168000 339750 60000 222000 181134 149500 126000 142000
206300 275000 109008 195400 85400 79900 122500 212000 116000 90350
555000 162900 199900 119500 188000 256000 161000 263435 62383 188700
124000 178740 146500 187000 440000 251000 132500 208900 380000 297000

```



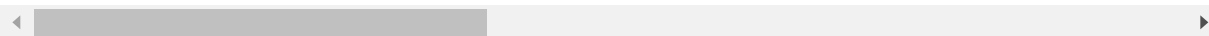
In [7]:

```
# dataset with numerical variables
numerics = ['int16', 'int32', 'int64', 'float16', 'float32', 'float64']
Data_numeric = Data_Frame.select_dtypes(include=numerics)
Data_numeric
```

Out[7]:

|      | Id   | MSSubClass | LotFrontage | LotArea | OverallQual | OverallCond | YearBuilt | YearRemodA |
|------|------|------------|-------------|---------|-------------|-------------|-----------|------------|
| 0    | 1    | 60         | 65.0        | 8450    | 7           | 5           | 2003      | 20         |
| 1    | 2    | 20         | 80.0        | 9600    | 6           | 8           | 1976      | 10         |
| 2    | 3    | 60         | 68.0        | 11250   | 7           | 5           | 2001      | 20         |
| 3    | 4    | 70         | 60.0        | 9550    | 7           | 5           | 1915      | 10         |
| 4    | 5    | 60         | 84.0        | 14260   | 8           | 5           | 2000      | 20         |
| ...  | ...  | ...        | ...         | ...     | ...         | ...         | ...       | ...        |
| 1455 | 1456 | 60         | 62.0        | 7917    | 6           | 5           | 1999      | 20         |
| 1456 | 1457 | 20         | 85.0        | 13175   | 6           | 6           | 1978      | 10         |
| 1457 | 1458 | 70         | 66.0        | 9042    | 7           | 9           | 1941      | 20         |
| 1458 | 1459 | 20         | 68.0        | 9717    | 5           | 6           | 1950      | 10         |
| 1459 | 1460 | 20         | 75.0        | 9937    | 5           | 6           | 1965      | 10         |

1460 rows × 38 columns



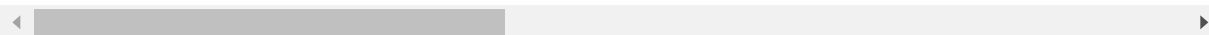
In [8]:

```
# dataset with categorical variables
Data_Category = Data_Frame.select_dtypes(include=['object'])
Data_Category
```

Out[8]:

|      | MSZoning | Street | Alley | LotShape | LandContour | Utilities | LotConfig | LandSlope | Neighbor |
|------|----------|--------|-------|----------|-------------|-----------|-----------|-----------|----------|
| 0    | RL       | Pave   | NaN   | Reg      | Lvl         | AllPub    | Inside    | Gtl       | Cr       |
| 1    | RL       | Pave   | NaN   | Reg      | Lvl         | AllPub    | FR2       | Gtl       | Ve       |
| 2    | RL       | Pave   | NaN   | IR1      | Lvl         | AllPub    | Inside    | Gtl       | Cr       |
| 3    | RL       | Pave   | NaN   | IR1      | Lvl         | AllPub    | Corner    | Gtl       | Cr       |
| 4    | RL       | Pave   | NaN   | IR1      | Lvl         | AllPub    | FR2       | Gtl       | No       |
| ...  | ...      | ...    | ...   | ...      | ...         | ...       | ...       | ...       | ...      |
| 1455 | RL       | Pave   | NaN   | Reg      | Lvl         | AllPub    | Inside    | Gtl       | Cr       |
| 1456 | RL       | Pave   | NaN   | Reg      | Lvl         | AllPub    | Inside    | Gtl       | NW       |
| 1457 | RL       | Pave   | NaN   | Reg      | Lvl         | AllPub    | Inside    | Gtl       | Cr       |
| 1458 | RL       | Pave   | NaN   | Reg      | Lvl         | AllPub    | Inside    | Gtl       | Cr       |
| 1459 | RL       | Pave   | NaN   | Reg      | Lvl         | AllPub    | Inside    | Gtl       | Ed       |

1460 rows × 43 columns



1. EDA of numerical variables:

a. Missing value treatment

In [9]:

```
Data_numeric_null = Data_numeric.isnull().sum()  
Data_numeric_null
```

Out[9]:

```
Id                0  
MSSubClass        0  
LotFrontage      259  
LotArea          0  
OverallQual      0  
OverallCond      0  
YearBuilt        0  
YearRemodAdd     0  
MasVnrArea       8  
BsmtFinSF1       0  
BsmtFinSF2       0  
BsmtUnfSF        0  
TotalBsmtSF      0  
1stFlrSF         0  
2ndFlrSF         0  
LowQualFinSF     0  
GrLivArea        0  
BsmtFullBath     0  
BsmtHalfBath     0  
FullBath         0  
HalfBath         0  
BedroomAbvGr     0  
KitchenAbvGr     0  
TotRmsAbvGrd     0  
Fireplaces       0  
GarageYrBlt      81  
GarageCars       0  
GarageArea       0  
WoodDeckSF       0  
OpenPorchSF      0  
EnclosedPorch    0  
3SsnPorch        0  
ScreenPorch      0  
PoolArea         0  
MiscVal          0  
MoSold           0  
YrSold           0  
SalePrice        0  
dtype: int64
```

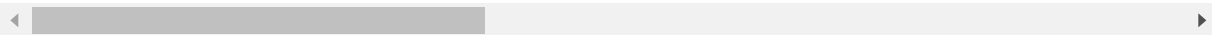
In [10]:

```
Data_numeric["LotFrontage"].fillna(Data_numeric["LotFrontage"].mean(),inplace=True)  
Data_numeric
```

Out[10]:

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

1460 rows × 38 columns





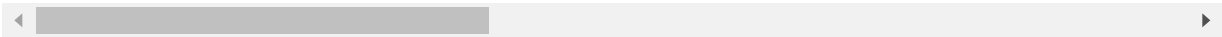
In [11]:

```
Data_numeric["GarageYrBlt"].fillna(Data_numeric["GarageYrBlt"].mean(),inplace=True)
Data_numeric
```

Out[11]:

|      | Id   | MSSubClass | LotFrontage | LotArea | OverallQual | OverallCond | YearBuilt | YearRemodA |
|------|------|------------|-------------|---------|-------------|-------------|-----------|------------|
| 0    | 1    | 60         | 65.0        | 8450    | 7           | 5           | 2003      | 20         |
| 1    | 2    | 20         | 80.0        | 9600    | 6           | 8           | 1976      | 10         |
| 2    | 3    | 60         | 68.0        | 11250   | 7           | 5           | 2001      | 20         |
| 3    | 4    | 70         | 60.0        | 9550    | 7           | 5           | 1915      | 10         |
| 4    | 5    | 60         | 84.0        | 14260   | 8           | 5           | 2000      | 20         |
| ...  | ...  | ...        | ...         | ...     | ...         | ...         | ...       | ...        |
| 1455 | 1456 | 60         | 62.0        | 7917    | 6           | 5           | 1999      | 20         |
| 1456 | 1457 | 20         | 85.0        | 13175   | 6           | 6           | 1978      | 10         |
| 1457 | 1458 | 70         | 66.0        | 9042    | 7           | 9           | 1941      | 20         |
| 1458 | 1459 | 20         | 68.0        | 9717    | 5           | 6           | 1950      | 10         |
| 1459 | 1460 | 20         | 75.0        | 9937    | 5           | 6           | 1965      | 10         |

1460 rows × 38 columns



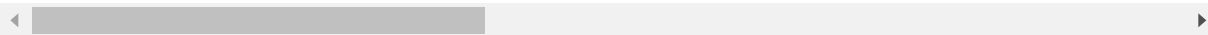
In [12]:

```
Data_numeric["MasVnrArea"].fillna(Data_numeric["MasVnrArea"].mean(),inplace=True)
Data_numeric
```

Out[12]:

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

1460 rows × 38 columns



In [13]:

```
#check any null value
Data_numeric.isna().any().sum()
```

Out[13]:

0

b. Identify the skewness and distribution

In [14]:

```
Data_Skew = Data_numeric.skew()  
Data_Skew
```

Out[14]:

|               |           |
|---------------|-----------|
| Id            | 0.000000  |
| MSSubClass    | 1.407657  |
| LotFrontage   | 2.384950  |
| LotArea       | 12.207688 |
| OverallQual   | 0.216944  |
| OverallCond   | 0.693067  |
| YearBuilt     | -0.613461 |
| YearRemodAdd  | -0.503562 |
| MasVnrArea    | 2.676412  |
| BsmtFinSF1    | 1.685503  |
| BsmtFinSF2    | 4.255261  |
| BsmtUnfSF     | 0.920268  |
| TotalBsmtSF   | 1.524255  |
| 1stFlrSF      | 1.376757  |
| 2ndFlrSF      | 0.813030  |
| LowQualFinSF  | 9.011341  |
| GrLivArea     | 1.366560  |
| BsmtFullBath  | 0.596067  |
| BsmtHalfBath  | 4.103403  |
| FullBath      | 0.036562  |
| HalfBath      | 0.675897  |
| BedroomAbvGr  | 0.211790  |
| KitchenAbvGr  | 4.488397  |
| TotRmsAbvGrd  | 0.676341  |
| Fireplaces    | 0.649565  |
| GarageYrBlt   | -0.668175 |
| GarageCars    | -0.342549 |
| GarageArea    | 0.179981  |
| WoodDeckSF    | 1.541376  |
| OpenPorchSF   | 2.364342  |
| EnclosedPorch | 3.089872  |
| 3SsnPorch     | 10.304342 |
| ScreenPorch   | 4.122214  |
| PoolArea      | 14.828374 |
| MiscVal       | 24.476794 |
| MoSold        | 0.212053  |
| YrSold        | 0.096269  |
| SalePrice     | 1.882876  |

dtype: float64

In [15]:

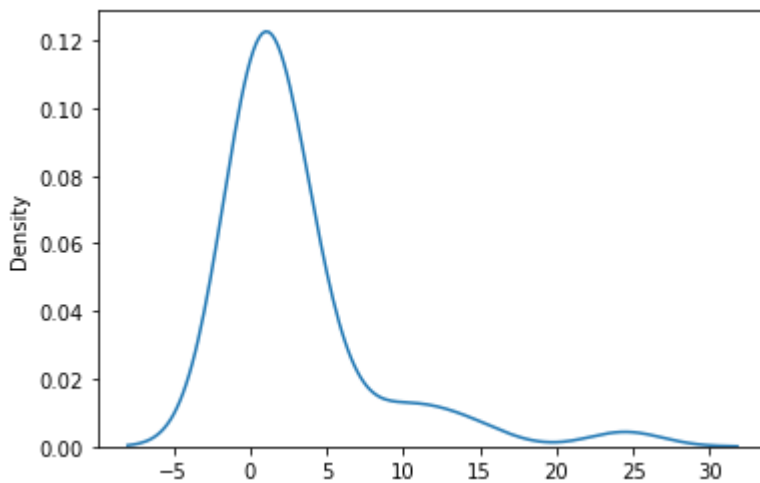
```
# Distributaion
import seaborn as sns
sns.distplot(Data_Skew,hist=False)
```

C:\Users\Lenovo\anaconda3\lib\site-packages\seaborn\distributions.py:2619: FutureWarning: `distplot` is a deprecated function and will be removed in a future version. Please adapt your code to use either `displot` (a figure-level function with similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).

warnings.warn(msg, FutureWarning)

Out[15]:

<AxesSubplot:ylabel='Density'>



c. Identify significant variables using a correlation matrix

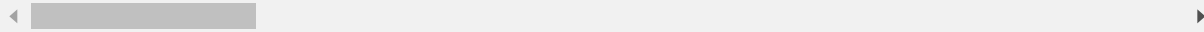
In [16]:

```
Data_numeric_corr = Data_numeric.corr()  
Data_numeric_corr  
Data_numeric_corr.style.background_gradient(cmap='coolwarm')
```

Out[16]:

|               | Id        | MSSubClass | LotFrontage | LotArea   | OverallQual | OverallCond | YearB   |
|---------------|-----------|------------|-------------|-----------|-------------|-------------|---------|
| Id            | 1.000000  | 0.011156   | -0.009601   | -0.033226 | -0.028365   | 0.012609    | -0.0127 |
| MSSubClass    | 0.011156  | 1.000000   | -0.357056   | -0.139781 | 0.032628    | -0.059316   | 0.0271  |
| LotFrontage   | -0.009601 | -0.357056  | 1.000000    | 0.306795  | 0.234196    | -0.052820   | 0.1171  |
| LotArea       | -0.033226 | -0.139781  | 0.306795    | 1.000000  | 0.105806    | -0.005636   | 0.0141  |
| OverallQual   | -0.028365 | 0.032628   | 0.234196    | 0.105806  | 1.000000    | -0.091932   | 0.5721  |
| OverallCond   | 0.012609  | -0.059316  | -0.052820   | -0.005636 | -0.091932   | 1.000000    | -0.3751 |
| YearBuilt     | -0.012713 | 0.027850   | 0.117598    | 0.014228  | 0.572323    | -0.375983   | 1.0000  |
| YearRemodAdd  | -0.021998 | 0.040581   | 0.082746    | 0.013788  | 0.550684    | 0.073741    | 0.5921  |
| MasVnrArea    | -0.050199 | 0.022895   | 0.179283    | 0.103960  | 0.410238    | -0.127788   | 0.3141  |
| BsmtFinSF1    | -0.005024 | -0.069836  | 0.215828    | 0.214103  | 0.239666    | -0.046231   | 0.2491  |
| BsmtFinSF2    | -0.005968 | -0.065649  | 0.043340    | 0.111170  | -0.059119   | 0.040229    | -0.0491 |
| BsmtUnfSF     | -0.007940 | -0.140759  | 0.122156    | -0.002618 | 0.308159    | -0.136841   | 0.1491  |
| TotalBsmtSF   | -0.015415 | -0.238518  | 0.363358    | 0.260833  | 0.537808    | -0.171098   | 0.3911  |
| 1stFlrSF      | 0.010496  | -0.251758  | 0.414266    | 0.299475  | 0.476224    | -0.144203   | 0.2811  |
| 2ndFlrSF      | 0.005590  | 0.307886   | 0.072483    | 0.050986  | 0.295493    | 0.028942    | 0.0101  |
| LowQualFinSF  | -0.044230 | 0.046474   | 0.036849    | 0.004779  | -0.030429   | 0.025494    | -0.1831 |
| GrLivArea     | 0.008273  | 0.074853   | 0.368392    | 0.263116  | 0.593007    | -0.079686   | 0.1991  |
| BsmtFullBath  | 0.002289  | 0.003491   | 0.091481    | 0.158155  | 0.111098    | -0.054942   | 0.1871  |
| BsmtHalfBath  | -0.020155 | -0.002333  | -0.006419   | 0.048046  | -0.040150   | 0.117821    | -0.0381 |
| FullBath      | 0.005587  | 0.131608   | 0.180424    | 0.126031  | 0.550600    | -0.194149   | 0.4681  |
| HalfBath      | 0.006784  | 0.177354   | 0.048258    | 0.014259  | 0.273458    | -0.060769   | 0.2421  |
| BedroomAbvGr  | 0.037719  | -0.023438  | 0.237023    | 0.119690  | 0.101676    | 0.012980    | -0.0701 |
| KitchenAbvGr  | 0.002951  | 0.281721   | -0.005805   | -0.017784 | -0.183882   | -0.087001   | -0.1741 |
| TotRmsAbvGrd  | 0.027239  | 0.040380   | 0.320146    | 0.190015  | 0.427452    | -0.057583   | 0.0951  |
| Fireplaces    | -0.019772 | -0.045569  | 0.235755    | 0.271364  | 0.396765    | -0.023820   | 0.1471  |
| GarageYrBlt   | 0.000070  | 0.080187   | 0.064324    | -0.024812 | 0.518018    | -0.306169   | 0.7801  |
| GarageCars    | 0.016570  | -0.040110  | 0.269729    | 0.154871  | 0.600671    | -0.185758   | 0.5371  |
| GarageArea    | 0.017634  | -0.098672  | 0.323663    | 0.180403  | 0.562022    | -0.151521   | 0.4781  |
| WoodDeckSF    | -0.029643 | -0.012579  | 0.077106    | 0.171698  | 0.238923    | -0.003334   | 0.2241  |
| OpenPorchSF   | -0.000477 | -0.006100  | 0.137454    | 0.084774  | 0.308819    | -0.032589   | 0.1881  |
| EnclosedPorch | 0.002889  | -0.012037  | 0.009790    | -0.018340 | -0.113937   | 0.070356    | -0.3871 |
| 3SsnPorch     | -0.046635 | -0.043825  | 0.062335    | 0.020423  | 0.030371    | 0.025504    | 0.0311  |
| ScreenPorch   | 0.001330  | -0.026030  | 0.037684    | 0.043160  | 0.064886    | 0.054811    | -0.0501 |
| PoolArea      | 0.057044  | 0.008283   | 0.180868    | 0.077672  | 0.065166    | -0.001985   | 0.0041  |

|                  | Id        | MSSubClass | LotFrontage | LotArea   | OverallQual | OverallCond | YearB  |
|------------------|-----------|------------|-------------|-----------|-------------|-------------|--------|
| <b>MiscVal</b>   | -0.006242 | -0.007683  | 0.001168    | 0.038068  | -0.031406   | 0.068777    | -0.034 |
| <b>MoSold</b>    | 0.021172  | -0.013585  | 0.010158    | 0.001205  | 0.070815    | -0.003511   | 0.012  |
| <b>YrSold</b>    | 0.000712  | -0.021407  | 0.006768    | -0.014261 | -0.027347   | 0.043950    | -0.013 |
| <b>SalePrice</b> | -0.021917 | -0.084284  | 0.334901    | 0.263843  | 0.790982    | -0.077856   | 0.522  |



Variables which have  $\geq 0.5$  correction with Saleprice

1. OverallQual (Overall material and finish quality)
2. YearBuilt (Original construction date)
3. YearRemodAdd (Remodel date)
4. TotalBsmtSF (Total square feet of basement area)
5. 1stFlrSF (First Floor square feet)
6. GrLivArea (Above grade (ground) living area square feet)
7. FullBath (Full bathrooms above grade)
8. TotRmsAbvGrd (Total rooms above grade (does not include bathrooms))
9. GarageCars (Size of the garage in car capacity)
10. GarageArea (Size of the garage in square feet)

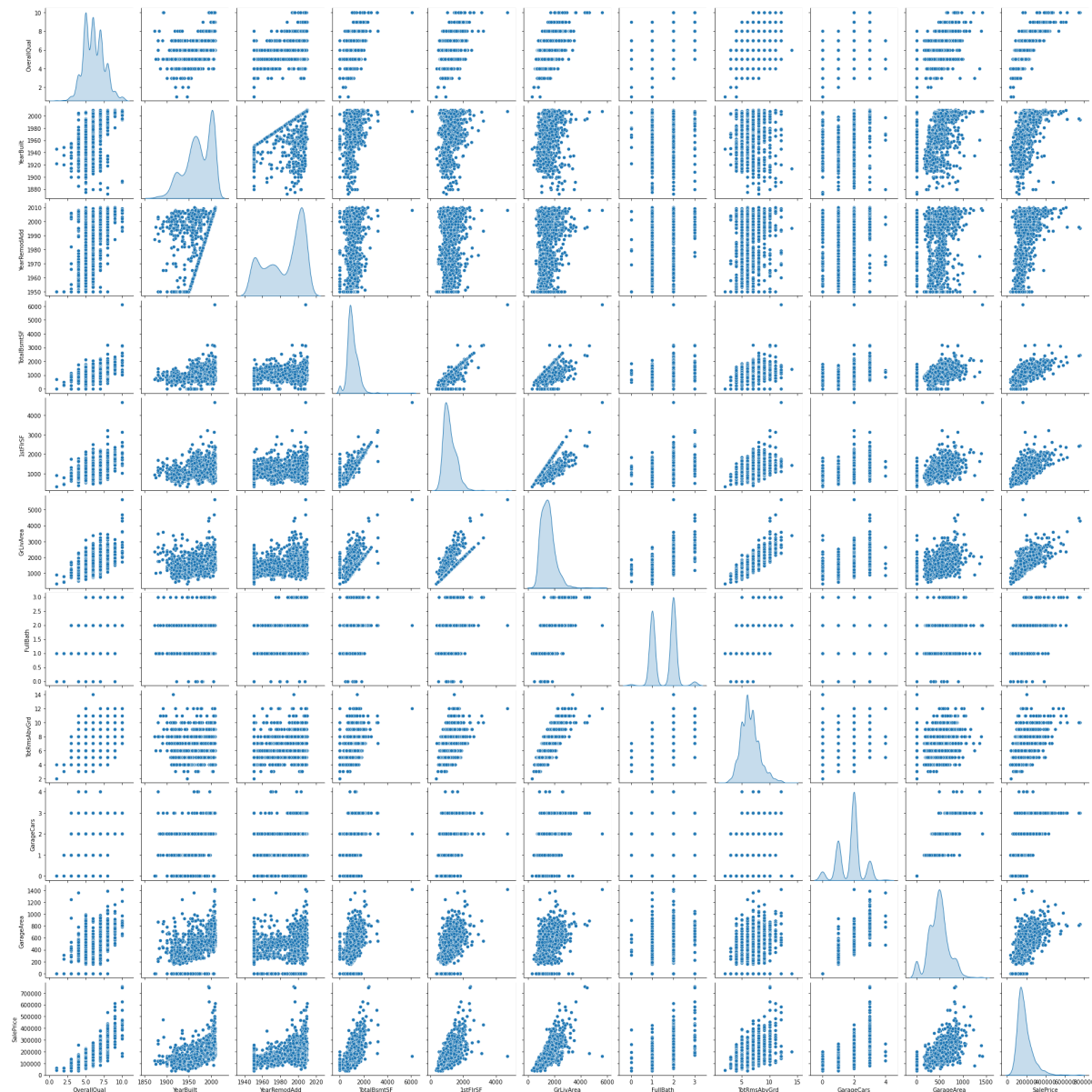
d. Pair plot for distribution and density

In [17]:

```
sns.pairplot(Data_numeric, vars=['OverallQual', 'YearBuilt', 'YearRemodAdd', 'TotalBsmtSF', '1stFlrSF', 'GrLivArea', 'FullBath', 'TotRmsAbvGrd', 'GarageCars', 'GarageArea', 'SalePrice'], diag_kind="kde")
```

Out[17]:

<seaborn.axisgrid.PairGrid at 0x2086d745be0>



As we can see in pairplot many variables match themselves

1. YearRemodAdd with SalePrice has a high resemblance to that of YearBuilt with SalePrice.
2. 1stFlrSF with SalePrice has a high resemblance to that of TotalBsmtSF with SalePrice.
3. TotRmsAbvGrd is highly correlated to GrLivArea.
4. GarageArea is highly correlated to GarageCars.

So, creating new numeric data which affect Saleprice



In [18]:

```
Data_numeric_new = Data_numeric[['OverallQual', 'YearBuilt', 'TotalBsmtSF', 'GrLivArea',  
'FullBath', 'GarageCars', 'SalePrice']]  
Data_numeric_new
```

Out[18]:

|      | OverallQual | YearBuilt | TotalBsmtSF | GrLivArea | FullBath | GarageCars | SalePrice |
|------|-------------|-----------|-------------|-----------|----------|------------|-----------|
| 0    | 7           | 2003      | 856         | 1710      | 2        | 2          | 208500    |
| 1    | 6           | 1976      | 1262        | 1262      | 2        | 2          | 181500    |
| 2    | 7           | 2001      | 920         | 1786      | 2        | 2          | 223500    |
| 3    | 7           | 1915      | 756         | 1717      | 1        | 3          | 140000    |
| 4    | 8           | 2000      | 1145        | 2198      | 2        | 3          | 250000    |
| ...  | ...         | ...       | ...         | ...       | ...      | ...        | ...       |
| 1455 | 6           | 1999      | 953         | 1647      | 2        | 2          | 175000    |
| 1456 | 6           | 1978      | 1542        | 2073      | 2        | 2          | 210000    |
| 1457 | 7           | 1941      | 1152        | 2340      | 2        | 1          | 266500    |
| 1458 | 5           | 1950      | 1078        | 1078      | 1        | 1          | 142125    |
| 1459 | 5           | 1965      | 1256        | 1256      | 1        | 1          | 147500    |

1460 rows × 7 columns

## 1. EDA of categorical variables

### a. Missing value treatment

In [19]:

```
Data_Category.isna().sum()
```

Out[19]:

|               |      |
|---------------|------|
| 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    |
| Function1     | 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

As we can see the data with larger null value is not related to sale price we can drop them. So, we can drop Alley, PoolQC, Fence, MiscFeature and FireplaceQu

In [20]:

```
Data_Category = Data_Category.drop(['Alley', 'PoolQC', 'Fence', 'MiscFeature', 'FireplaceQu'],  
axis=1)
```

In [21]:

```
Data_Category.isna().sum()
```

Out[21]:

|               |    |
|---------------|----|
| MSZoning      | 0  |
| Street        | 0  |
| 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  |
| Function1     | 0  |
| GarageType    | 81 |
| GarageFinish  | 81 |
| GarageQual    | 81 |
| GarageCond    | 81 |
| PavedDrive    | 0  |
| SaleType      | 0  |
| SaleCondition | 0  |

dtype: int64

As we can see still we have null values but they are very less number we can drop the same too.

In [22]:

```
Data_Category.dropna(inplace=True)
```

In [23]:

```
Data_Category.isna().sum()
```

Out[23]:

|               |   |
|---------------|---|
| MSZoning      | 0 |
| Street        | 0 |
| 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    | 0 |
| ExterQual     | 0 |
| ExterCond     | 0 |
| Foundation    | 0 |
| BsmtQual      | 0 |
| BsmtCond      | 0 |
| BsmtExposure  | 0 |
| BsmtFinType1  | 0 |
| BsmtFinType2  | 0 |
| Heating       | 0 |
| HeatingQC     | 0 |
| CentralAir    | 0 |
| Electrical    | 0 |
| KitchenQual   | 0 |
| Function1     | 0 |
| GarageType    | 0 |
| GarageFinish  | 0 |
| GarageQual    | 0 |
| GarageCond    | 0 |
| PavedDrive    | 0 |
| SaleType      | 0 |
| SaleCondition | 0 |

dtype: int64

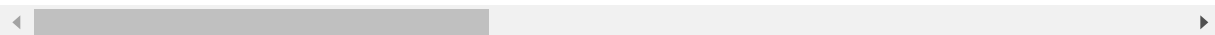
In [24]:

```
Data_Category
```

Out[24]:

|      | MSZoning | Street | LotShape | LandContour | Utilities | LotConfig | LandSlope | Neighborhood |
|------|----------|--------|----------|-------------|-----------|-----------|-----------|--------------|
| 0    | RL       | Pave   | Reg      | Lvl         | AllPub    | Inside    | Gtl       | CollgCr      |
| 1    | RL       | Pave   | Reg      | Lvl         | AllPub    | FR2       | Gtl       | Veenker      |
| 2    | RL       | Pave   | IR1      | Lvl         | AllPub    | Inside    | Gtl       | CollgCr      |
| 3    | RL       | Pave   | IR1      | Lvl         | AllPub    | Corner    | Gtl       | Crawfor      |
| 4    | RL       | Pave   | IR1      | Lvl         | AllPub    | FR2       | Gtl       | NoRidge      |
| ...  | ...      | ...    | ...      | ...         | ...       | ...       | ...       | ...          |
| 1455 | RL       | Pave   | Reg      | Lvl         | AllPub    | Inside    | Gtl       | Gilbert      |
| 1456 | RL       | Pave   | Reg      | Lvl         | AllPub    | Inside    | Gtl       | NWAmes       |
| 1457 | RL       | Pave   | Reg      | Lvl         | AllPub    | Inside    | Gtl       | Crawfor      |
| 1458 | RL       | Pave   | Reg      | Lvl         | AllPub    | Inside    | Gtl       | mes          |
| 1459 | RL       | Pave   | Reg      | Lvl         | AllPub    | Inside    | Gtl       | Edwards      |

1338 rows × 38 columns



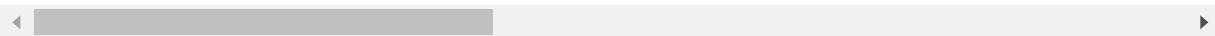
In [25]:

```
#Adding SalePrice to the category data frame  
Data_Category['SalePrice'] = Data_Frame.loc[Data_Category.index, 'SalePrice'].copy()  
Data_Category.head(2)
```

Out[25]:

|   | MSZoning | Street | LotShape | LandContour | Utilities | LotConfig | LandSlope | Neighborhood | Co |
|---|----------|--------|----------|-------------|-----------|-----------|-----------|--------------|----|
| 0 | RL       | Pave   | Reg      | Lvl         | AllPub    | Inside    | Gtl       | CollgCr      |    |
| 1 | RL       | Pave   | Reg      | Lvl         | AllPub    | FR2       | Gtl       | Veenker      |    |

2 rows × 39 columns

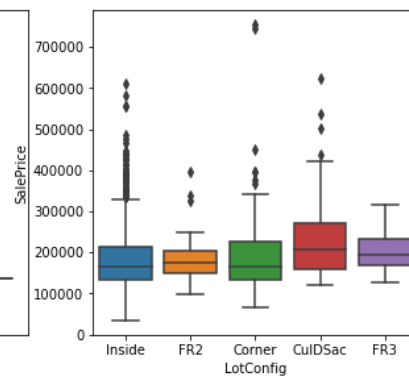
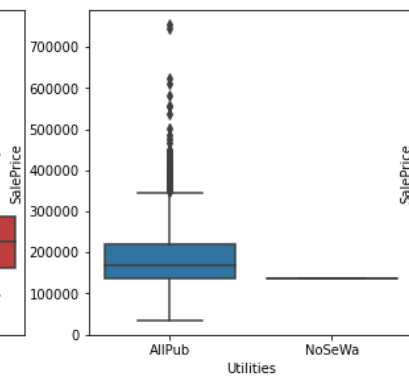
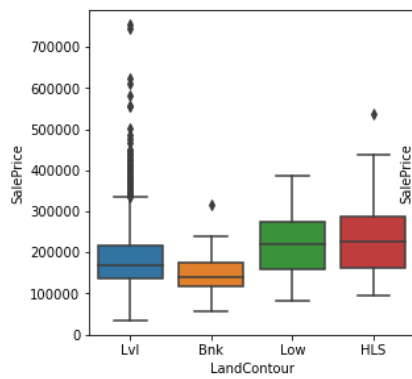
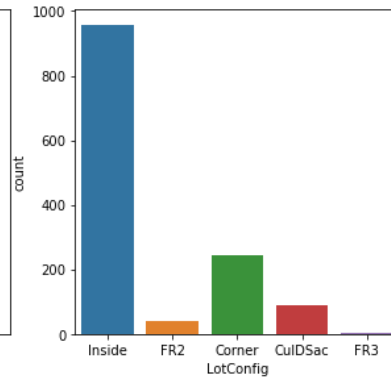
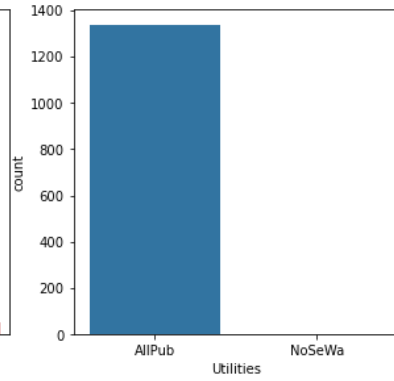
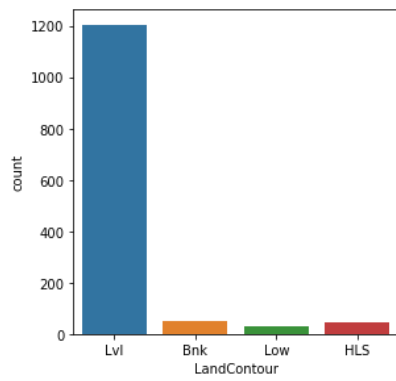
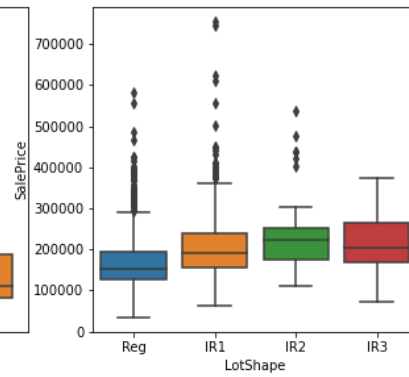
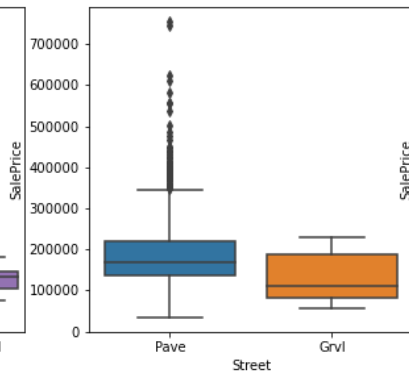
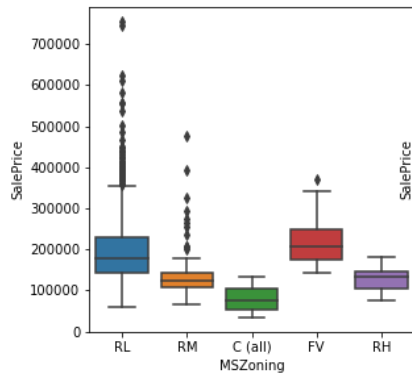
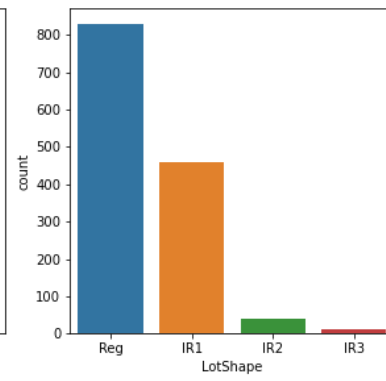
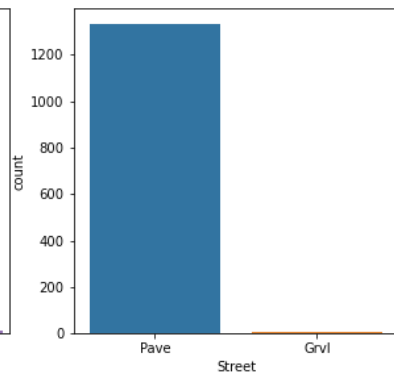
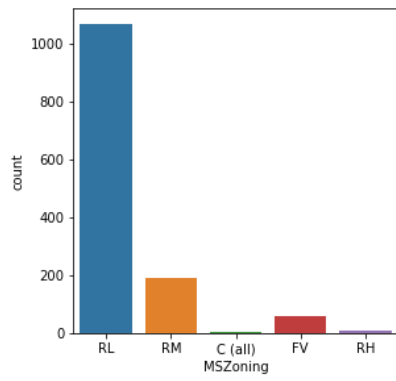


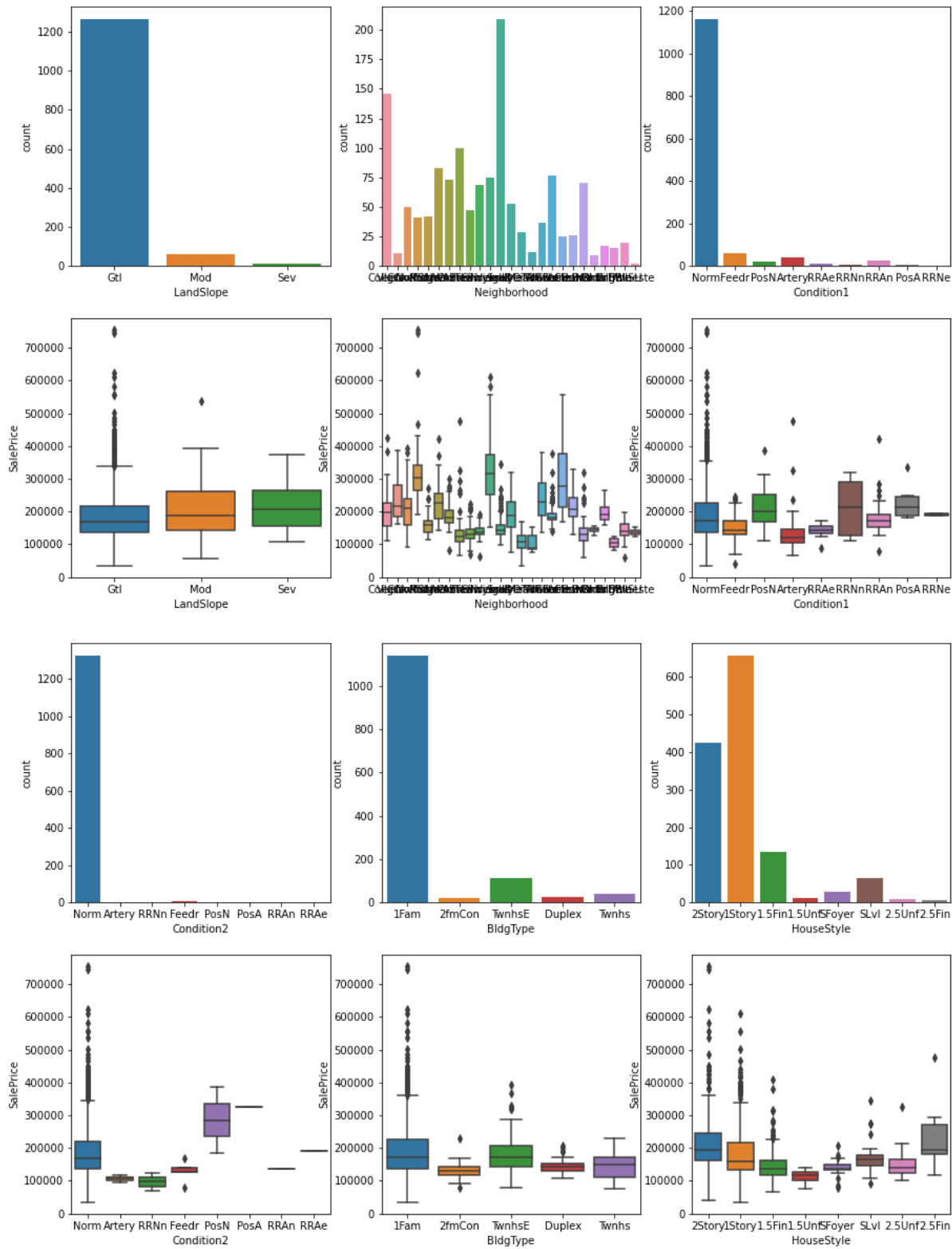
b. Count plot and box plot for bivariate analysis

In [27]:

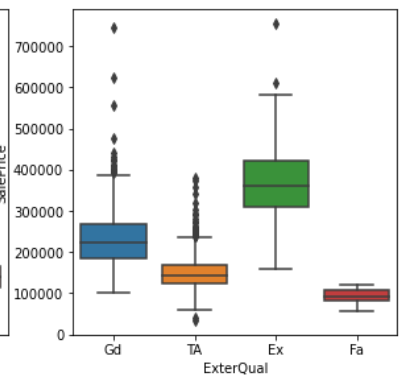
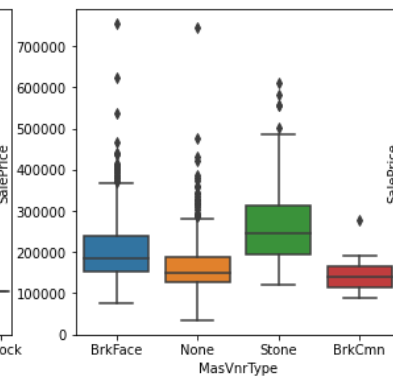
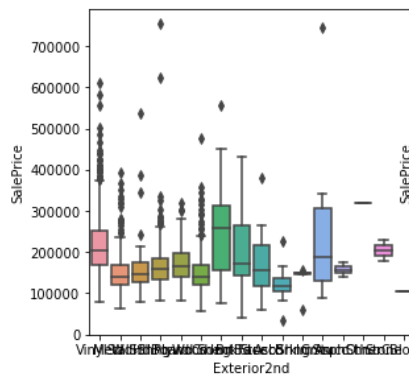
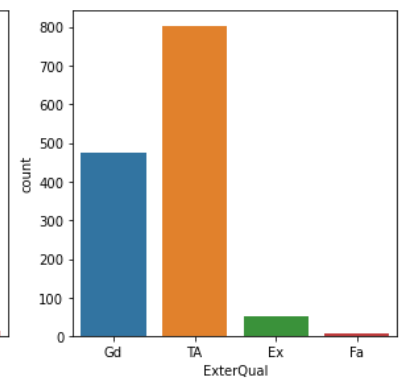
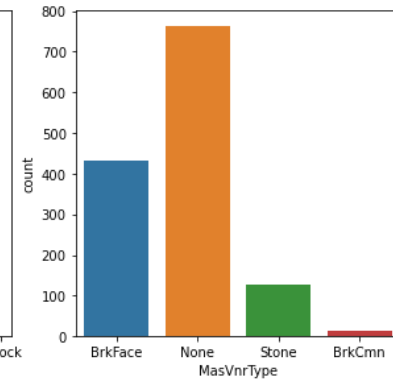
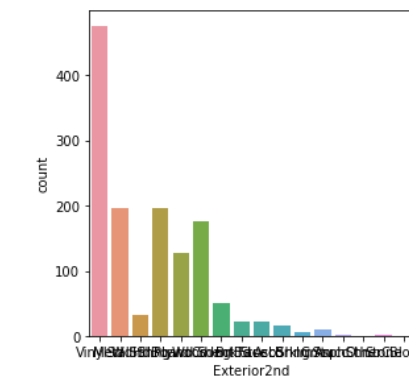
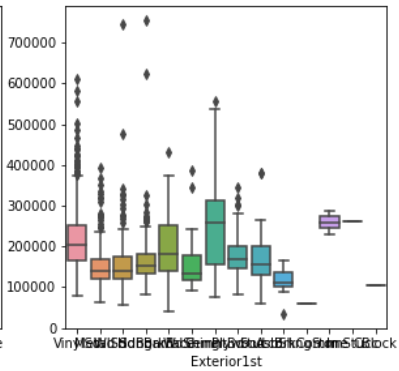
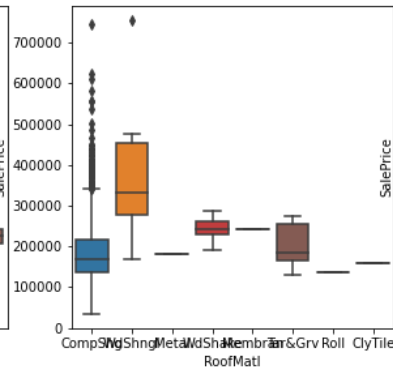
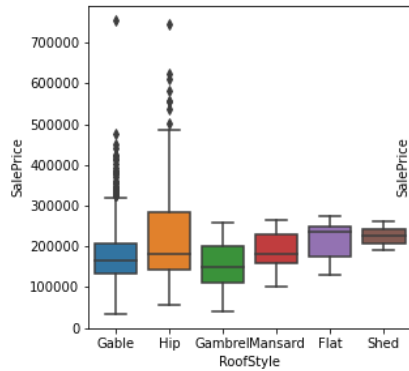
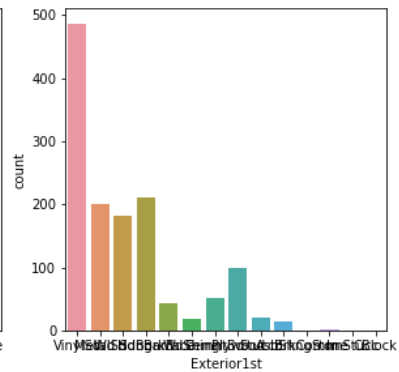
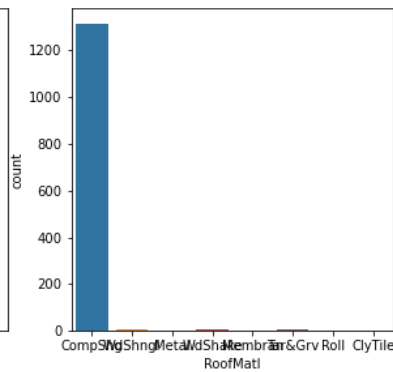
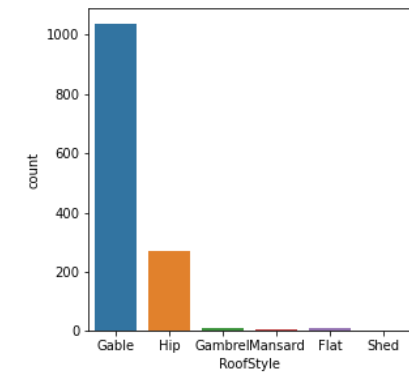
```
i = 1
fig = plt.figure(figsize = (15,10))
for c in list(Data_Category.columns):
    if i <= 3:
        if c != 'SalePrice':
            ax1 = fig.add_subplot(2,3,i)
            sns.countplot(data = Data_Category, x=c, ax = ax1) #For countplot
            ax2 = fig.add_subplot(2,3,i+3)
            sns.boxplot(data = Data_Category, x=c, y='SalePrice', ax=ax2) #For boxplot

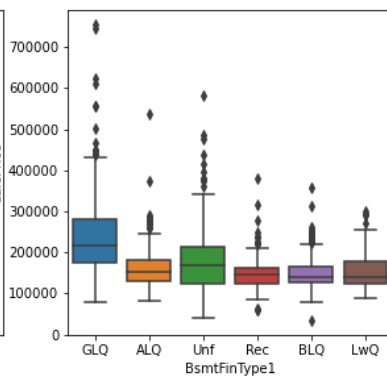
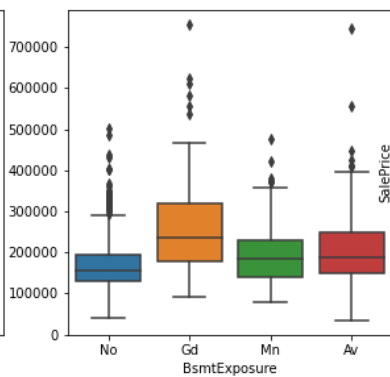
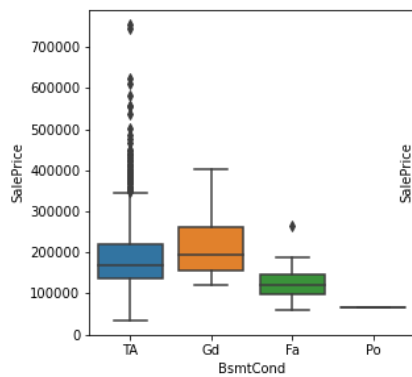
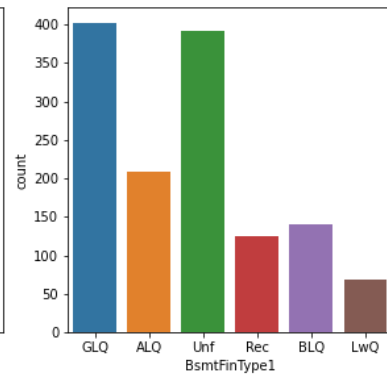
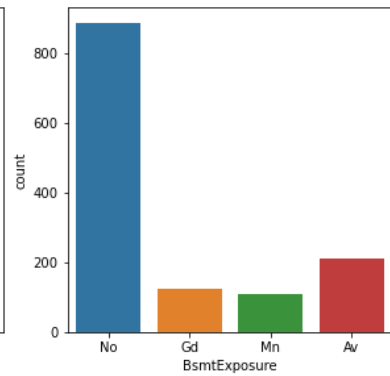
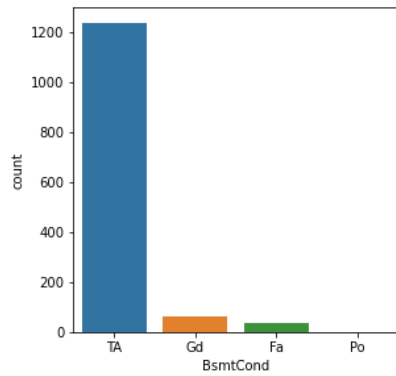
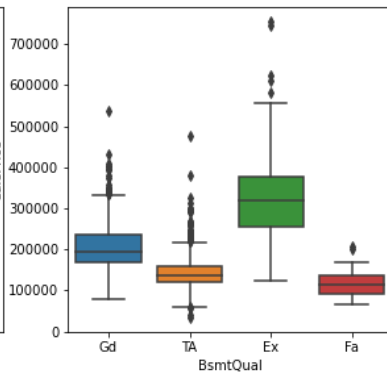
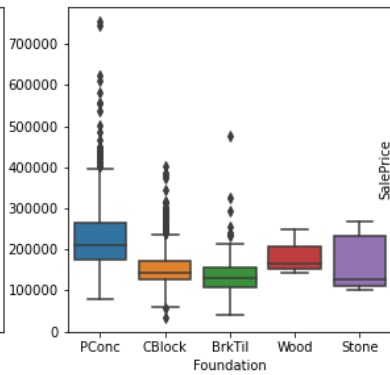
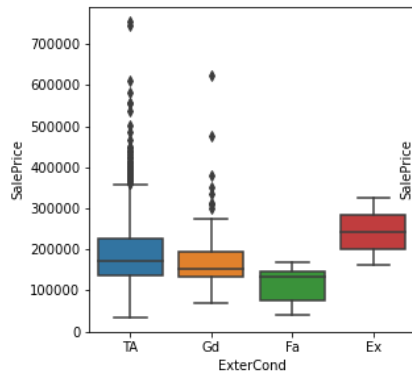
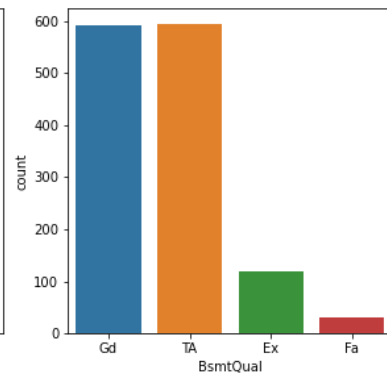
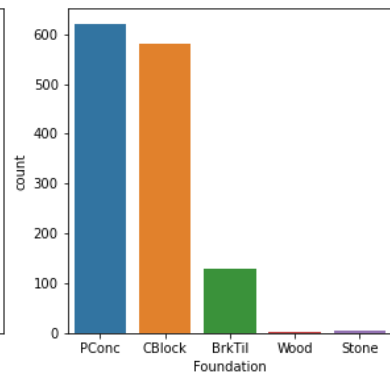
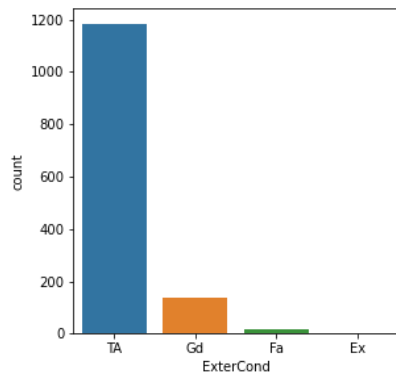
    i = i +1
if i == 4:
    fig = plt.figure(figsize = (15,10))
    i = 1
```

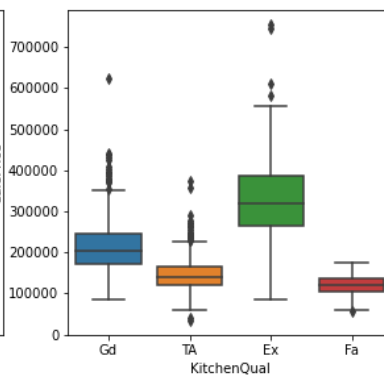
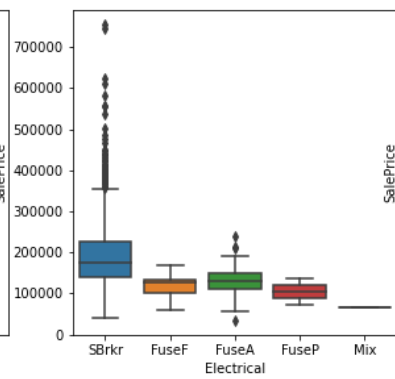
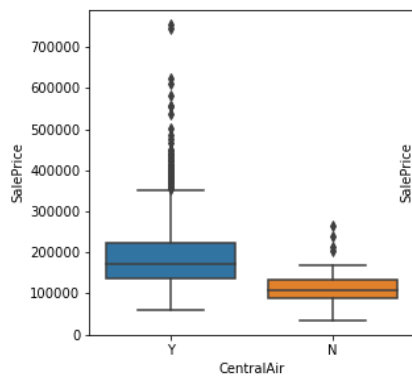
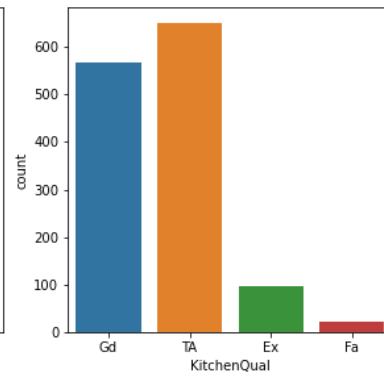
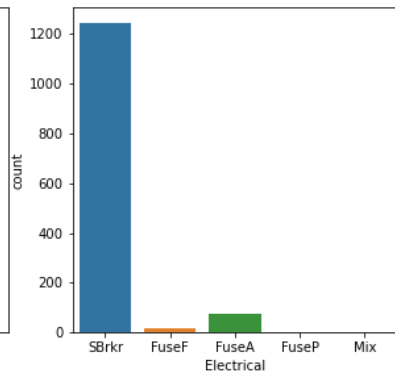
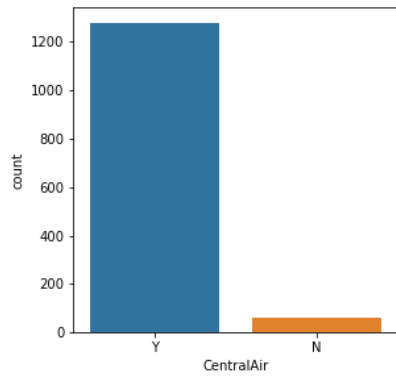
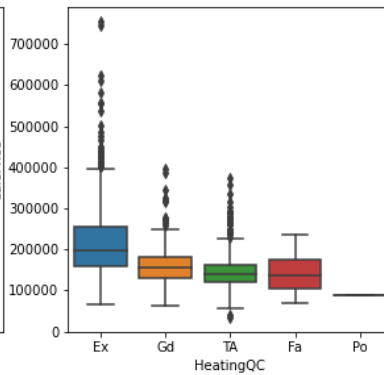
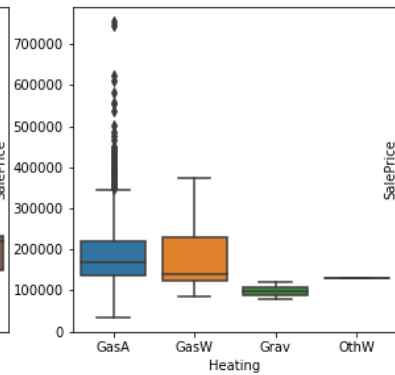
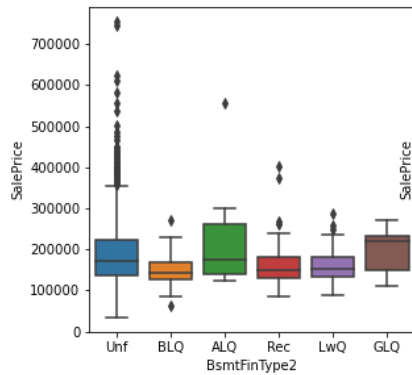
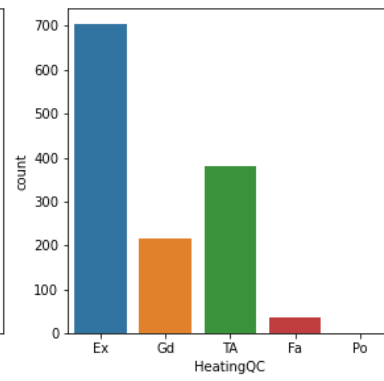
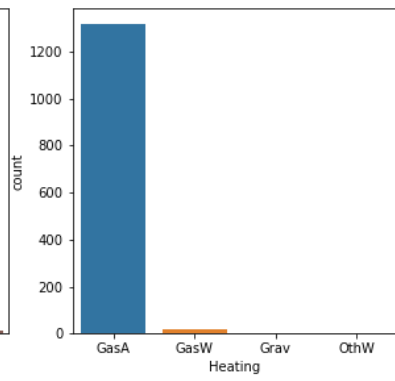
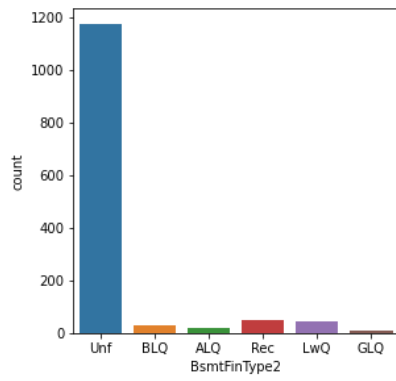


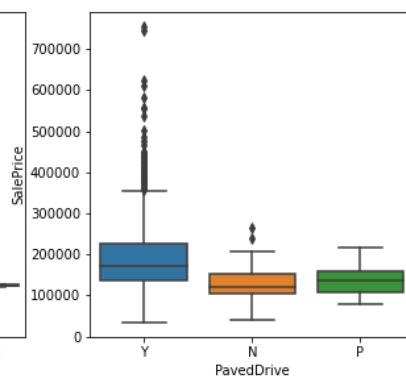
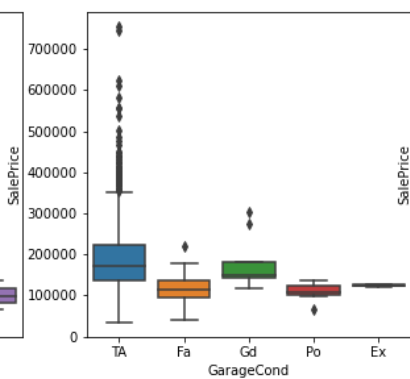
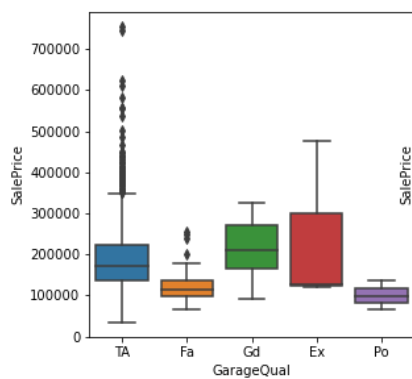
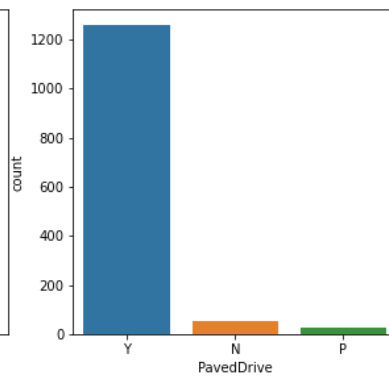
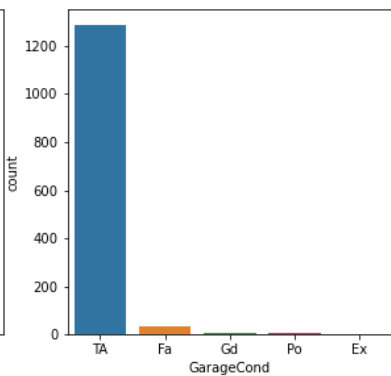
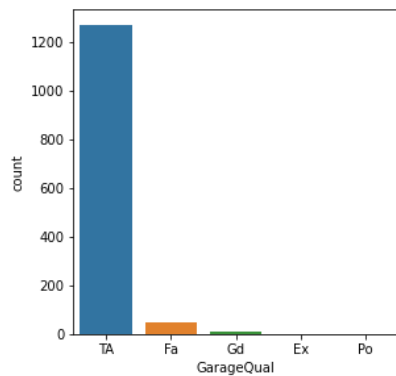
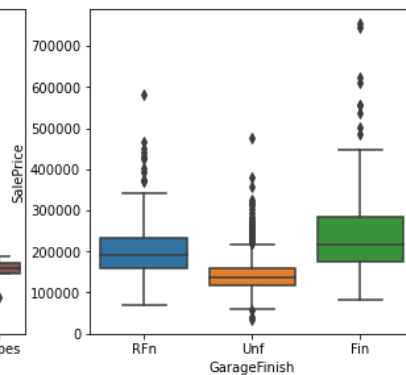
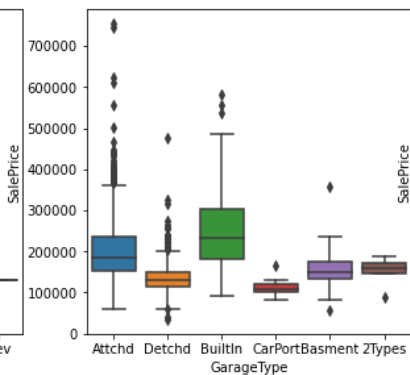
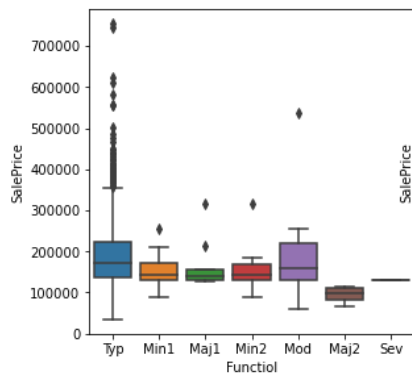
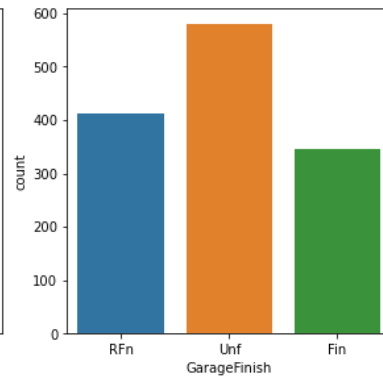
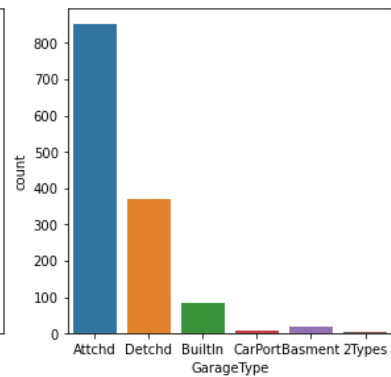
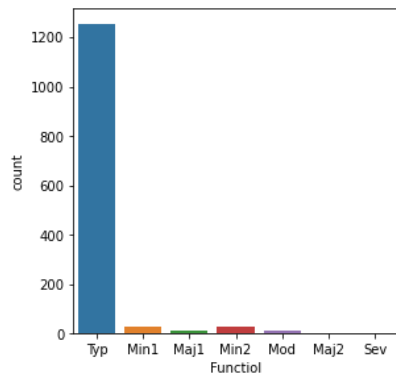


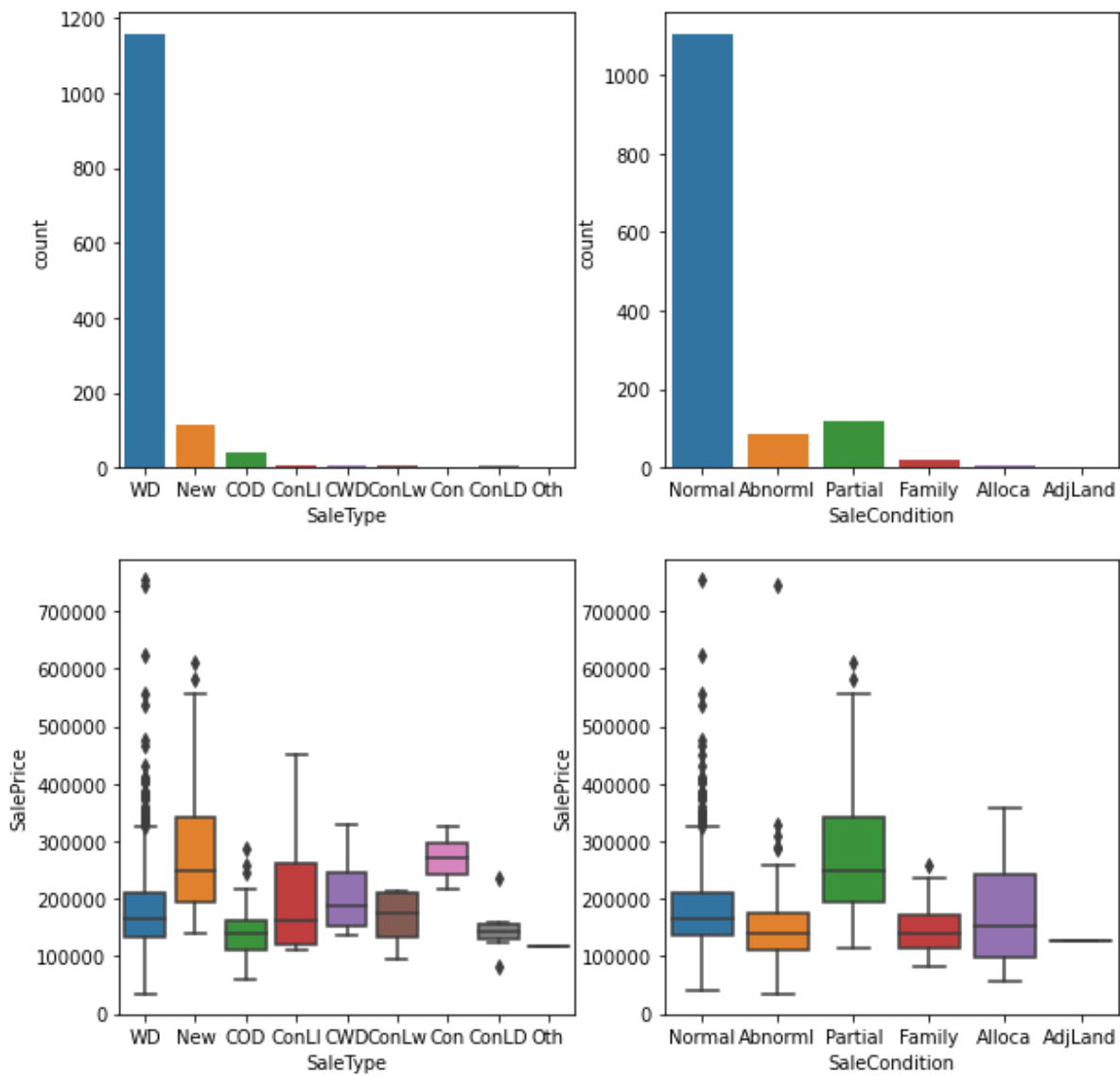












<Figure size 1080x720 with 0 Axes>

c. Identify significant variables using p-values and Chi-Square values

In [28]:

```
import numpy as np
import scipy.stats as stats
from scipy.stats import chi2_contingency
class ChiSquare:
    #Function to determine p-value and perform chi-square test
    def __init__(self, dataframe):
        self.df = dataframe
        self.p = None #P-Value
        self.chi2 = None #Chi-square Test Statistic
        self.dof = None

        self.dfObserved = None
        self.dfExpected = None

    #Function to print the results of p-value and chi-square test
    def _print_chisquare_result(self, colX, alpha):
        result = ""
        if self.p < alpha:
            result = "{0} is IMPORTANT for Prediction".format(colX)
        else:
            result = "{0} is NOT an important predictor. (Discard {0} from model)".format(colX)
        print(result)

    #Function to determine chi-square and p-value Less than or equal to 0.05
    def TestIndependence(self, colX, colY, alpha=0.05):
        X = self.df[colX].astype(str)
        Y = self.df[colY].astype(str)

        self.dfObserved = pd.crosstab(Y, X)
        chi2, p, dof, expected = stats.chi2_contingency(self.dfObserved.values)
        self.p = p
        self.chi2 = chi2
        self.dof = dof

        self.dfExpected = pd.DataFrame(expected, columns=self.dfObserved.columns, index =
self.dfObserved.index)

        self._print_chisquare_result(colX, alpha)

#Initializing ChiSquare Class
cT = ChiSquare(Data_Category)
#Perform Feature Selection
testColumns = ['MSZoning', 'Street', 'LotShape', 'LandContour', 'Utilities', 'LotConfig',
'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'BldgType', 'HouseStyle', 'RoofSt
yle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType', 'ExterQual', 'ExterCond', 'F
oundation', 'BsmtQual', 'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinType2', 'Heati
ng', 'HeatingQC', 'CentralAir', 'Electrical', 'KitchenQual', 'Functiol', 'GarageType', 'Ga
rageFinish', 'GarageQual', 'GarageCond', 'PavedDrive', 'SaleType', 'SaleCondition', 'SalePr
ice']
for var in testColumns:
    cT.TestIndependence(colX=var, colY="SalePrice" )
```

MSZoning is IMPORTANT for Prediction  
Street is IMPORTANT for Prediction  
LotShape is IMPORTANT for Prediction  
LandContour is NOT an important predictor. (Discard LandContour from model)  
Utilities is NOT an important predictor. (Discard Utilities from model)  
LotConfig is NOT an important predictor. (Discard LotConfig from model)  
LandSlope is NOT an important predictor. (Discard LandSlope from model)  
Neighborhood is IMPORTANT for Prediction  
Condition1 is NOT an important predictor. (Discard Condition1 from model)  
Condition2 is IMPORTANT for Prediction  
BldgType is NOT an important predictor. (Discard BldgType from model)  
HouseStyle is NOT an important predictor. (Discard HouseStyle from model)  
RoofStyle is NOT an important predictor. (Discard RoofStyle from model)  
RoofMatl is NOT an important predictor. (Discard RoofMatl from model)  
Exterior1st is NOT an important predictor. (Discard Exterior1st from model)  
Exterior2nd is NOT an important predictor. (Discard Exterior2nd from model)  
MasVnrType is IMPORTANT for Prediction  
ExterQual is IMPORTANT for Prediction  
ExterCond is NOT an important predictor. (Discard ExterCond from model)  
Foundation is IMPORTANT for Prediction  
BsmtQual is IMPORTANT for Prediction  
BsmtCond is IMPORTANT for Prediction  
BsmtExposure is IMPORTANT for Prediction  
BsmtFinType1 is NOT an important predictor. (Discard BsmtFinType1 from model)  
BsmtFinType2 is NOT an important predictor. (Discard BsmtFinType2 from model)  
Heating is NOT an important predictor. (Discard Heating from model)  
HeatingQC is NOT an important predictor. (Discard HeatingQC from model)  
CentralAir is IMPORTANT for Prediction  
Electrical is IMPORTANT for Prediction  
KitchenQual is IMPORTANT for Prediction  
Functio1 is NOT an important predictor. (Discard Functio1 from model)  
GarageType is IMPORTANT for Prediction  
GarageFinish is IMPORTANT for Prediction  
GarageQual is IMPORTANT for Prediction  
GarageCond is NOT an important predictor. (Discard GarageCond from model)  
PavedDrive is NOT an important predictor. (Discard PavedDrive from model)  
SaleType is IMPORTANT for Prediction  
SaleCondition is IMPORTANT for Prediction  
SalePrice is IMPORTANT for Prediction

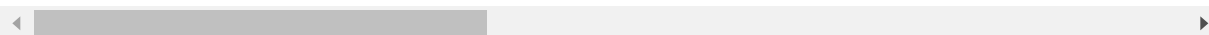
In [30]:

```
Data_Category = Data_Category[['MSZoning', 'Street', 'LotShape', 'Neighborhood', 'Condition2', 'MasVnrType', 'ExterQual', 'Foundation', 'BsmtQual', 'BsmtCond', 'BsmtExposure', 'CentralAir', 'Electrical', 'KitchenQual', 'GarageType', 'GarageFinish', 'GarageQual', 'SaleType', 'SaleCondition', 'SalePrice']]
Data_Category
```

Out[30]:

|      | MSZoning | Street | LotShape | Neighborhood | Condition2 | MasVnrType | ExterQual | Foundation |
|------|----------|--------|----------|--------------|------------|------------|-----------|------------|
| 0    | RL       | Pave   | Reg      | CollgCr      | Norm       | BrkFace    | Gd        | PCon       |
| 1    | RL       | Pave   | Reg      | Veenker      | Norm       | None       | TA        | CBloc      |
| 2    | RL       | Pave   | IR1      | CollgCr      | Norm       | BrkFace    | Gd        | PCon       |
| 3    | RL       | Pave   | IR1      | Crawfor      | Norm       | None       | TA        | Brk1       |
| 4    | RL       | Pave   | IR1      | NoRidge      | Norm       | BrkFace    | Gd        | PCon       |
| ...  | ...      | ...    | ...      | ...          | ...        | ...        | ...       | .          |
| 1455 | RL       | Pave   | Reg      | Gilbert      | Norm       | None       | TA        | PCon       |
| 1456 | RL       | Pave   | Reg      | NWAmes       | Norm       | Stone      | TA        | CBloc      |
| 1457 | RL       | Pave   | Reg      | Crawfor      | Norm       | None       | Ex        | Ston       |
| 1458 | RL       | Pave   | Reg      | mes          | Norm       | None       | TA        | CBloc      |
| 1459 | RL       | Pave   | Reg      | Edwards      | Norm       | None       | Gd        | CBloc      |

1338 rows × 20 columns



1. Combine all the significant categorical and numerical variables



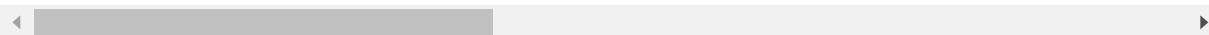
In [31]:

```
DataFrame_New = pd.merge(Data_Category , Data_numeric , how="outer", on=["SalePrice"])  
DataFrame_New
```

Out[31]:

|      | MSZoning | Street | LotShape | Neighborhood | Condition2 | MasVnrType | ExterQual | Foundatio |
|------|----------|--------|----------|--------------|------------|------------|-----------|-----------|
| 0    | RL       | Pave   | Reg      | CollgCr      | Norm       | BrkFace    | Gd        | PCon      |
| 1    | RL       | Pave   | Reg      | Veenker      | Norm       | None       | TA        | CBloc     |
| 2    | RL       | Pave   | IR1      | CollgCr      | Norm       | BrkFace    | Gd        | PCon      |
| 3    | RL       | Pave   | IR1      | CollgCr      | Norm       | BrkFace    | Gd        | PCon      |
| 4    | RL       | Pave   | Reg      | BrkSide      | Norm       | None       | TA        | CBloc     |
| ...  | ...      | ...    | ...      | ...          | ...        | ...        | ...       | .         |
| 6378 | NaN      | NaN    | NaN      | NaN          | NaN        | NaN        | NaN       | Na        |
| 6379 | NaN      | NaN    | NaN      | NaN          | NaN        | NaN        | NaN       | Na        |
| 6380 | NaN      | NaN    | NaN      | NaN          | NaN        | NaN        | NaN       | Na        |
| 6381 | NaN      | NaN    | NaN      | NaN          | NaN        | NaN        | NaN       | Na        |
| 6382 | NaN      | NaN    | NaN      | NaN          | NaN        | NaN        | NaN       | Na        |

6383 rows × 57 columns

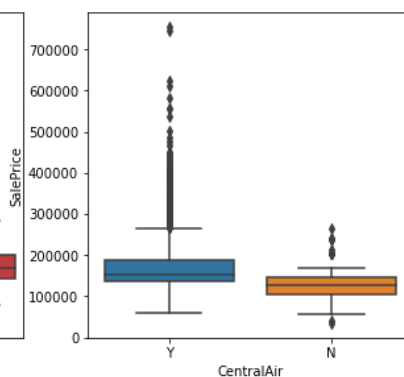
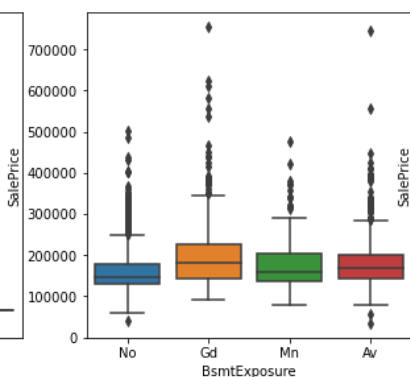
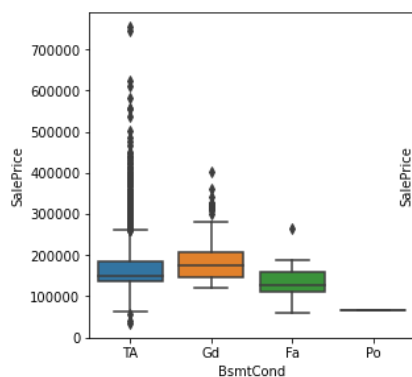
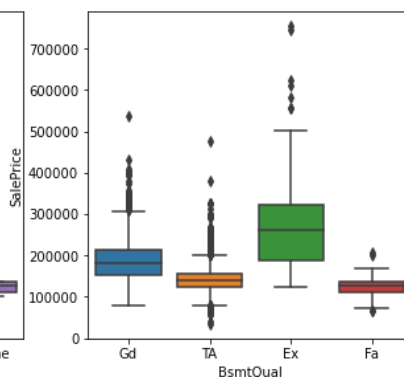
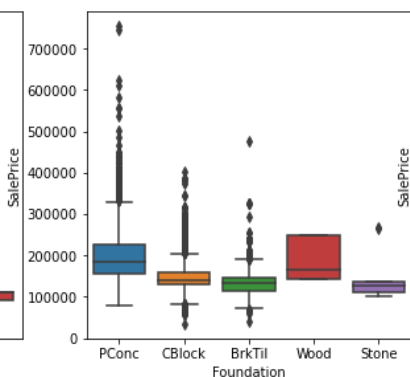
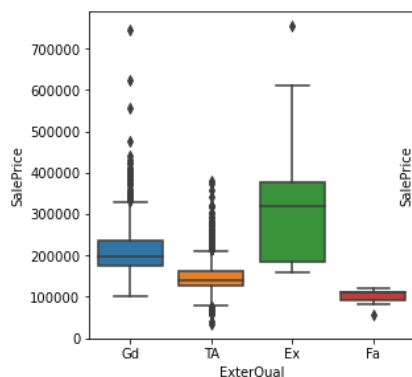
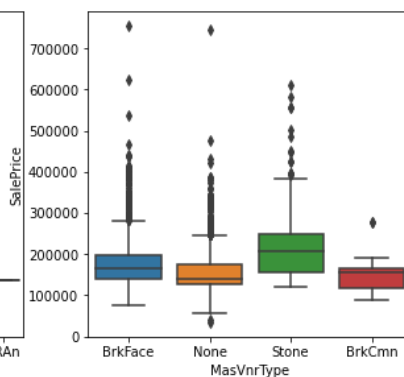
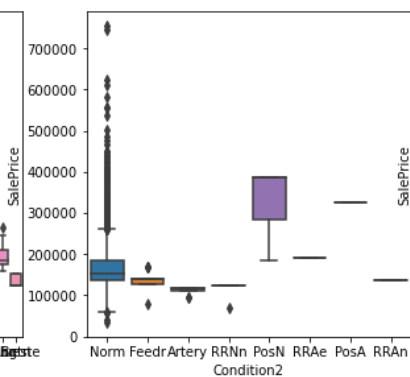
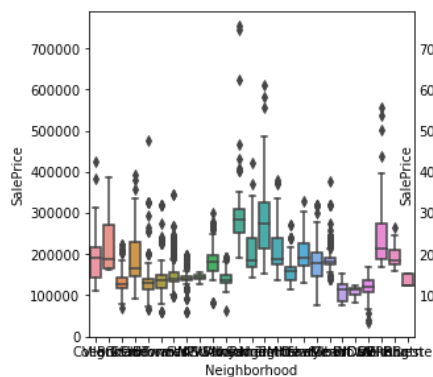
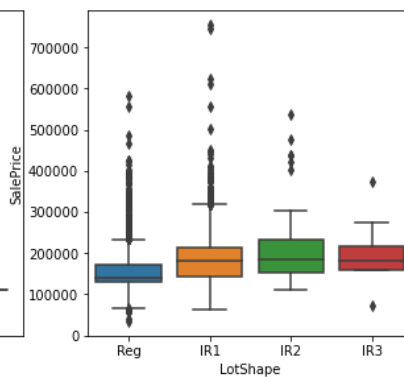
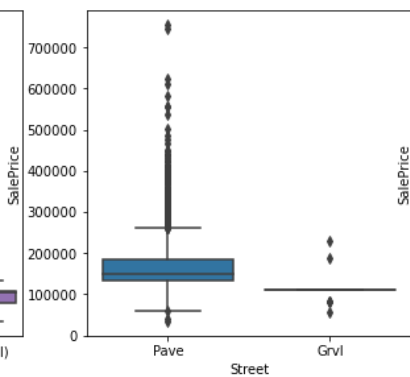
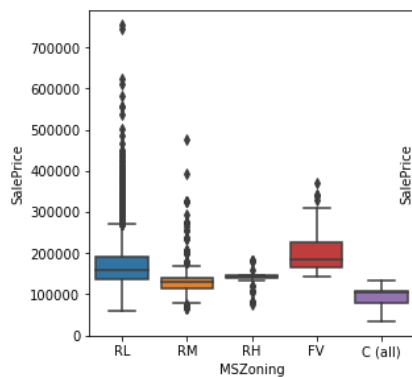


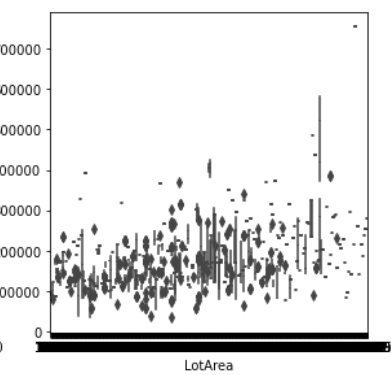
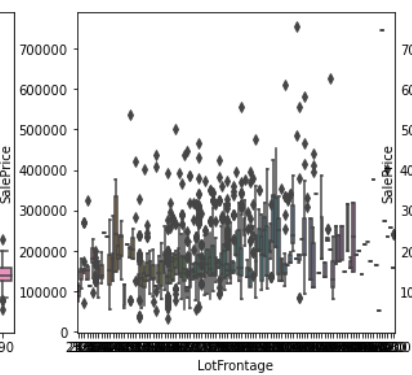
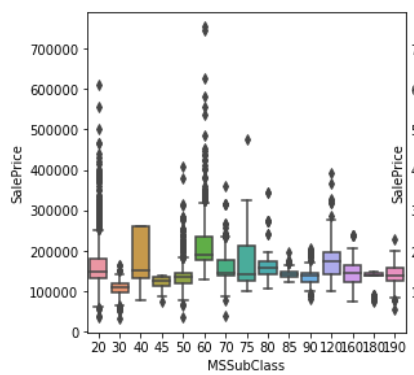
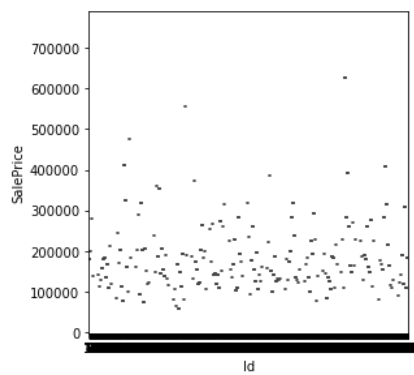
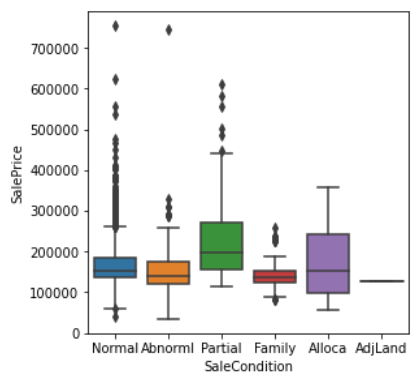
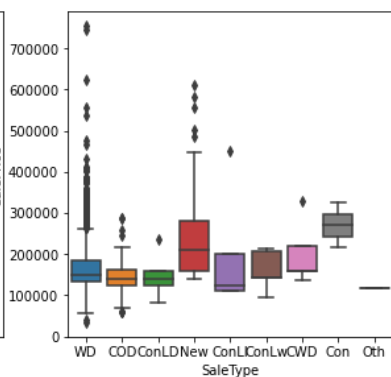
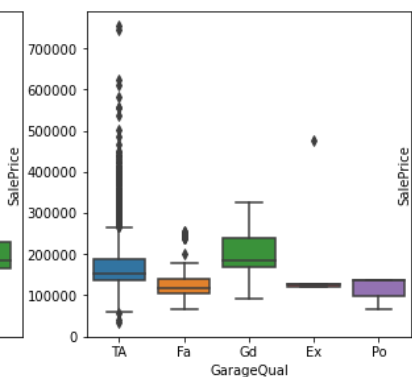
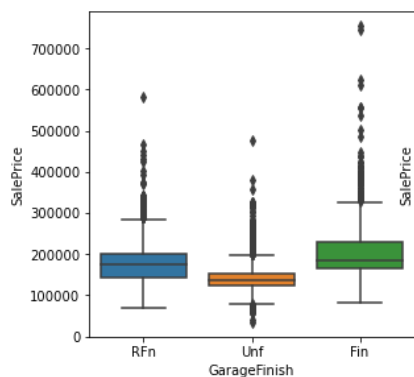
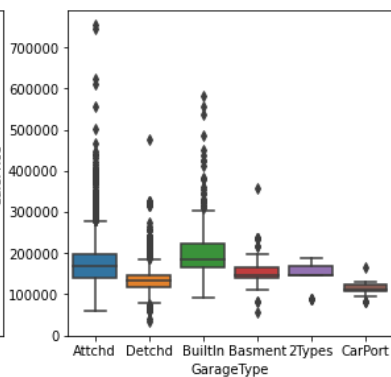
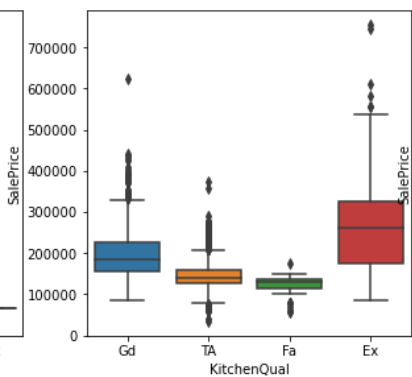
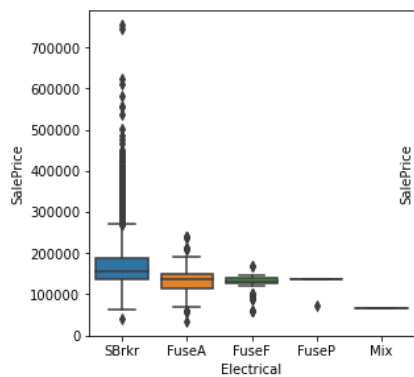
a. Plot box plot for the new dataset to find the variables with outliers

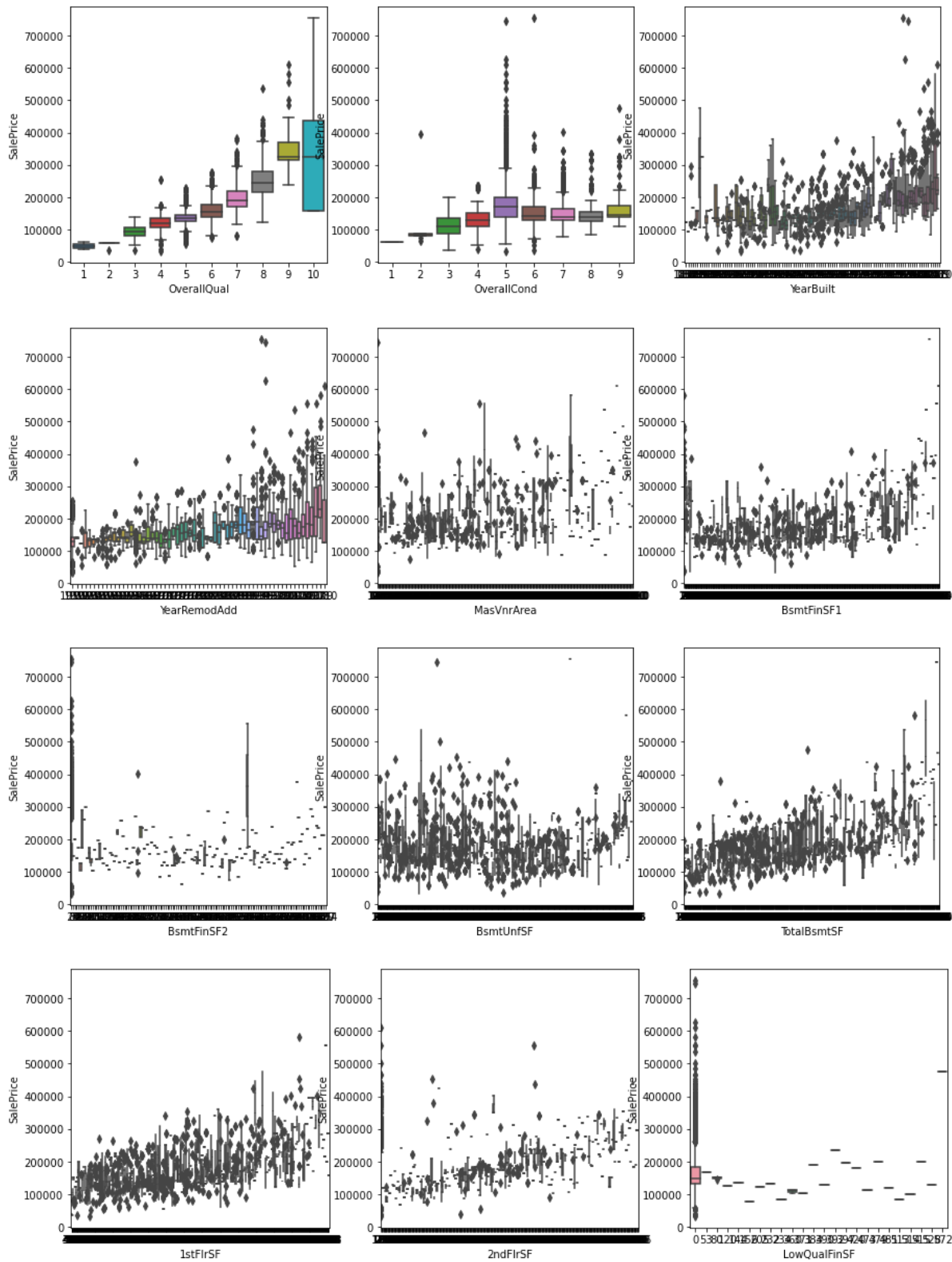
In [ ]:

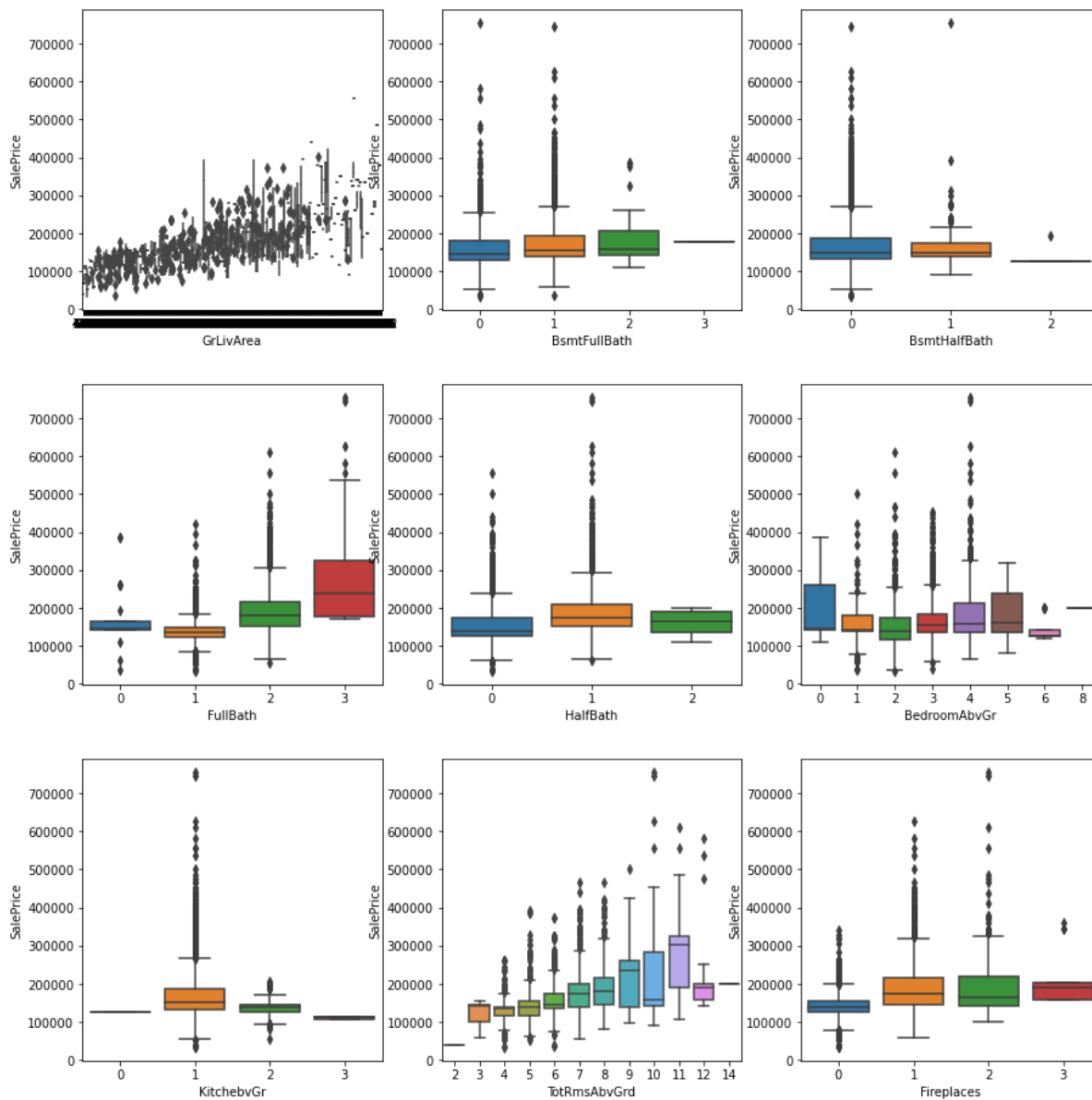
```
#Function to plot all independent categorical variables with SalePrice and count plot
i = 1
fig = plt.figure(figsize = (15,10))
for c in list(DataFrame_New.columns):
    if i <= 3:
        if c != 'SalePrice':
            ax2 = fig.add_subplot(2,3,i+3)
            sns.boxplot(data=DataFrame_New, x=c, y='SalePrice', ax=ax2) #for boxplot

    i = i +1
if i == 4:
    fig = plt.figure(figsize = (15,10))
    i =1
```









In [ ]: