In [1]:
```
import pandas as pd
```

1) Understand the dataset:

In [2]:
```
pd.read_csv('PEP1.csv')
```

Out[2]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContou
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lv
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lv
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lv
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lv
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lv
...	...	...	...	...	...	...	...	...	..
1455	1456	60	RL	62.0	7917	Pave	NaN	Reg	Lv
1456	1457	20	RL	85.0	13175	Pave	NaN	Reg	Lv
1457	1458	70	RL	66.0	9042	Pave	NaN	Reg	Lv
1458	1459	20	RL	68.0	9717	Pave	NaN	Reg	Lv
1459	1460	20	RL	75.0	9937	Pave	NaN	Reg	Lv

1460 rows × 81 columns

In [3]:
```
PEP=pd.read_csv('PEP1.csv')
```

1(a) Identify the shape of the dataset

In [4]:
```
PEP.shape
```

Out[4]:
```
(1460, 81)
```

In [5]:
```
len(PEP)
```

Out[5]:
```
1460
```

1(b) Identify variables with null values

In [6]:
```
PEP.isnull().sum().head(40)
```

Out[6]:

```
Id                0
MSSubClass        0
MSZoning          0
LotFrontage     259
LotArea           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
OverallQual       0
OverallCond       0
YearBuilt         0
YearRemodAdd      0
RoofStyle         0
RoofMatl          0
Exterior1st       0
Exterior2nd       0
MasVnrType        8
MasVnrArea        8
ExterQual         0
ExterCond         0
Foundation        0
BsmtQual         37
BsmtCond         37
BsmtExposure     38
BsmtFinType1     37
BsmtFinSF1        0
BsmtFinType2     38
BsmtFinSF2        0
BsmtUnfSF         0
TotalBsmtSF       0
Heating           0
dtype: int64
```

In [16]:

```
PEP.isnull().sum().tail(41)
```

Out[16]:
```
HeatingQC           0
CentralAir          0
Electrical          1
1stFlrSF            0
2ndFlrSF            0
LowQualFinSF        0
GrLivArea           0
BsmtFullBath        0
BsmtHalfBath        0
FullBath            0
HalfBath            0
BedroomAbvGr        0
KitchebvGr          0
KitchenQual         0
TotRmsAbvGrd        0
Functiol            0
Fireplaces          0
FireplaceQu       690
GarageType         81
GarageYrBlt        81
GarageFinish       81
GarageCars          0
GarageArea          0
GarageQual         81
GarageCond         81
PavedDrive          0
WoodDeckSF          0
OpenPorchSF         0
EnclosedPorch       0
3SsnPorch           0
ScreenPorch         0
PoolArea            0
PoolQC           1453
Fence            1179
MiscFeature      1406
MiscVal             0
MoSold              0
YrSold              0
SaleType            0
SaleCondition       0
SalePrice           0
dtype: int64
```

1(c) Identify variables with unique values

In [17]:
```
PEP.nunique().head(40)
```

Out[17]:

```
Id              1460
MSSubClass        15
MSZoning           5
LotFrontage      110
LotArea         1073
Street             2
Alley              2
LotShape           4
LandContour        4
Utilities          2
LotConfig          5
LandSlope          3
Neighborhood      25
Condition1         9
Condition2         8
BldgType           5
HouseStyle         8
OverallQual       10
OverallCond        9
YearBuilt        112
YearRemodAdd      61
RoofStyle          6
RoofMatl           8
Exterior1st       15
Exterior2nd       16
MasVnrType         4
MasVnrArea       327
ExterQual          4
ExterCond          5
Foundation         6
BsmtQual           4
BsmtCond           4
BsmtExposure       4
BsmtFinType1       6
BsmtFinSF1       637
BsmtFinType2       6
BsmtFinSF2       144
BsmtUnfSF        780
TotalBsmtSF      721
Heating            6
dtype: int64
```

In [18]:

```
PEP.nunique().tail(41)
```

Out[18]:
```
HeatingQC          5
CentralAir         2
Electrical         5
1stFlrSF         753
2ndFlrSF         417
LowQualFinSF      24
GrLivArea        861
BsmtFullBath       4
BsmtHalfBath       3
FullBath           4
HalfBath           3
BedroomAbvGr       8
KitchebvGr         4
KitchenQual        4
TotRmsAbvGrd      12
Functiol           7
Fireplaces         4
FireplaceQu        5
GarageType         6
GarageYrBlt       97
GarageFinish       3
GarageCars         5
GarageArea       441
GarageQual         5
GarageCond         5
PavedDrive         3
WoodDeckSF       274
OpenPorchSF      202
EnclosedPorch    120
3SsnPorch         20
ScreenPorch       76
PoolArea           8
PoolQC             3
Fence              4
MiscFeature        4
MiscVal           21
MoSold            12
YrSold             5
SaleType           9
SaleCondition      6
SalePrice        663
dtype: int64
```

2) Generate a separate dataset for numerical and categorical variables

In [19]:
```
categorical_data=PEP.select_dtypes(include='object')
print(categorical_data)
```

```
     MSZoning Street Alley LotShape LandContour Utilities LotConfig LandSlope  \
0          RL   Pave   NaN      Reg         Lvl    AllPub    Inside       Gtl
1          RL   Pave   NaN      Reg         Lvl    AllPub       FR2       Gtl
2          RL   Pave   NaN      IR1         Lvl    AllPub    Inside       Gtl
3          RL   Pave   NaN      IR1         Lvl    AllPub    Corner       Gtl
4          RL   Pave   NaN      IR1         Lvl    AllPub       FR2       Gtl
...       ...    ...   ...      ...         ...       ...       ...       ...
1455       RL   Pave   NaN      Reg         Lvl    AllPub    Inside       Gtl
1456       RL   Pave   NaN      Reg         Lvl    AllPub    Inside       Gtl
1457       RL   Pave   NaN      Reg         Lvl    AllPub    Inside       Gtl
1458       RL   Pave   NaN      Reg         Lvl    AllPub    Inside       Gtl
1459       RL   Pave   NaN      Reg         Lvl    AllPub    Inside       Gtl

     Neighborhood Condition1  ... GarageType GarageFinish GarageQual  \
0         CollgCr       Norm  ...     Attchd          RFn         TA
1         Veenker      Feedr  ...     Attchd          RFn         TA
2         CollgCr       Norm  ...     Attchd          RFn         TA
3         Crawfor       Norm  ...     Detchd          Unf         TA
4         NoRidge       Norm  ...     Attchd          RFn         TA
...           ...        ...  ...        ...          ...        ...
1455      Gilbert       Norm  ...     Attchd          RFn         TA
1456       NWAmes       Norm  ...     Attchd          Unf         TA
1457      Crawfor       Norm  ...     Attchd          RFn         TA
1458          mes       Norm  ...     Attchd          Unf         TA
1459      Edwards       Norm  ...     Attchd          Fin         TA

     GarageCond PavedDrive PoolQC  Fence MiscFeature SaleType SaleCondition
0            TA          Y    NaN    NaN         NaN       WD        Normal
1            TA          Y    NaN    NaN         NaN       WD        Normal
2            TA          Y    NaN    NaN         NaN       WD        Normal
3            TA          Y    NaN    NaN         NaN       WD       Abnorml
4            TA          Y    NaN    NaN         NaN       WD        Normal
...         ...        ...    ...    ...         ...      ...           ...
1455         TA          Y    NaN    NaN         NaN       WD        Normal
1456         TA          Y    NaN  MnPrv         NaN       WD        Normal
1457         TA          Y    NaN  GdPrv        Shed       WD        Normal
1458         TA          Y    NaN    NaN         NaN       WD        Normal
1459         TA          Y    NaN    NaN         NaN       WD        Normal

[1460 rows x 43 columns]
```

In [20]:
```
categorical_data.shape[1]
```

Out[20]:
```
43
```

In [21]:
```
numeric_data = PEP.select_dtypes(exclude='object')
print(numeric_data)
```

```
        Id  MSSubClass  LotFrontage  LotArea  OverallQual  OverallCond  \
0        1          60         65.0     8450            7            5
1        2          20         80.0     9600            6            8
2        3          60         68.0    11250            7            5
3        4          70         60.0     9550            7            5
4        5          60         84.0    14260            8            5
...    ...         ...          ...      ...          ...          ...
1455  1456          60         62.0     7917            6            5
1456  1457          20         85.0    13175            6            6
1457  1458          70         66.0     9042            7            9
1458  1459          20         68.0     9717            5            6
1459  1460          20         75.0     9937            5            6

      YearBuilt  YearRemodAdd  MasVnrArea  BsmtFinSF1  ...  WoodDeckSF  \
0          2003          2003       196.0         706  ...           0
1          1976          1976         0.0         978  ...         298
2          2001          2002       162.0         486  ...           0
3          1915          1970         0.0         216  ...           0
4          2000          2000       350.0         655  ...         192
...         ...           ...         ...         ...  ...         ...
1455       1999          2000         0.0           0  ...           0
1456       1978          1988       119.0         790  ...         349
1457       1941          2006         0.0         275  ...           0
1458       1950          1996         0.0          49  ...         366
1459       1965          1965         0.0         830  ...         736

      OpenPorchSF  EnclosedPorch  3SsnPorch  ScreenPorch  PoolArea  MiscVal  \
0              61              0          0            0         0        0
1               0              0          0            0         0        0
2              42              0          0            0         0        0
3              35            272          0            0         0        0
4              84              0          0            0         0        0
...           ...            ...        ...          ...       ...      ...
1455           40              0          0            0         0        0
1456            0              0          0            0         0        0
1457           60              0          0            0         0     2500
1458            0            112          0            0         0        0
1459           68              0          0            0         0        0

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

[1460 rows x 38 columns]
```

In [22]:
```
numeric_data.shape[1]
```

Out[22]:
```
38
```

3) EDA of numerical variables

3(a) MISSING VALUE TREATMENT FOR NUMERICAL VARIABLES

In [24]:
```
numeric_data.isna().sum()
```

Out[24]:
```
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
KitchebvGr       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 [25]:
```
numeric_data.isna().sum()[numeric_data.isna().sum()>0]
```

Out[25]:
```
LotFrontage    259
MasVnrArea       8
GarageYrBlt     81
dtype: int64
```

In [26]:
```
numeric_data.isna().sum()[numeric_data.isna().sum()>0].shape
```

Out[26]:
```
(3,)
```

In [361…
```
# Percentage of missing values in each numerical variable
```

In [27]:
```
percentage=(numeric_data.isna().sum()[numeric_data.isna().sum()>0]/1460)*100
print(percentage)
```

```
LotFrontage    17.739726
MasVnrArea      0.547945
GarageYrBlt     5.547945
dtype: float64
```

In [360…
```
# Removing the numerical variables having maximum missing values
```

In [28]:
```
No_miss_val = numeric_data.drop(['LotFrontage'],axis=1)
print(No_miss_val)
```

```
        Id  MSSubClass  LotArea  OverallQual  OverallCond  YearBuilt  \
0        1          60     8450            7            5       2003
1        2          20     9600            6            8       1976
2        3          60    11250            7            5       2001
3        4          70     9550            7            5       1915
4        5          60    14260            8            5       2000
...    ...         ...      ...          ...          ...        ...
1455  1456          60     7917            6            5       1999
1456  1457          20    13175            6            6       1978
1457  1458          70     9042            7            9       1941
1458  1459          20     9717            5            6       1950
1459  1460          20     9937            5            6       1965

      YearRemodAdd  MasVnrArea  BsmtFinSF1  BsmtFinSF2  ...  WoodDeckSF  \
0             2003       196.0         706           0  ...           0
1             1976         0.0         978           0  ...         298
2             2002       162.0         486           0  ...           0
3             1970         0.0         216           0  ...           0
4             2000       350.0         655           0  ...         192
...            ...         ...         ...         ...  ...         ...
1455          2000         0.0           0           0  ...           0
1456          1988       119.0         790         163  ...         349
1457          2006         0.0         275           0  ...           0
1458          1996         0.0          49        1029  ...         366
1459          1965         0.0         830         290  ...         736

      OpenPorchSF  EnclosedPorch  3SsnPorch  ScreenPorch  PoolArea  MiscVal  \
0              61              0          0            0         0        0
1               0              0          0            0         0        0
2              42              0          0            0         0        0
3              35            272          0            0         0        0
4              84              0          0            0         0        0
...           ...            ...        ...          ...       ...      ...
1455           40              0          0            0         0        0
1456            0              0          0            0         0        0
1457           60              0          0            0         0     2500
1458            0            112          0            0         0        0
1459           68              0          0            0         0        0

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

[1460 rows x 37 columns]
```

In [362…
```
# Percentage of missing values in each numerical variable after Removing the numer
```

In [29]:
```
No_miss_val.isna().sum()[No_miss_val.isna().sum()>0]/1460*100
```

Out[29]:
```
MasVnrArea     0.547945
GarageYrBlt    5.547945
dtype: float64
```

In [363…
```
# Missing Value imputation
```

In [75]:
```
PEP.loc[PEP.MasVnrArea.isna(),'MasVnrArea']
```

Out[75]:
```
234    NaN
529    NaN
650    NaN
936    NaN
973    NaN
977    NaN
1243   NaN
1278   NaN
Name: MasVnrArea, dtype: float64
```

In [89]:
```
PEP.MasVnrArea.median()
```

Out[89]:
```
0.0
```

In [95]:
```
data_imput_1 = PEP.fillna(PEP.MasVnrArea.median())
```

In [96]:
```
data_imput_1.loc[PEP.MasVnrArea.isna(),'MasVnrArea']
```

Out[96]:
```
234     0.0
529     0.0
650     0.0
936     0.0
973     0.0
977     0.0
1243    0.0
1278    0.0
Name: MasVnrArea, dtype: float64
```

In [79]:
```
PEP.loc[PEP.GarageYrBlt.isna(),'GarageYrBlt']
```

Out[79]:
```
39     NaN
48     NaN
78     NaN
88     NaN
89     NaN
       ..
1349   NaN
1407   NaN
1449   NaN
1450   NaN
1453   NaN
Name: GarageYrBlt, Length: 81, dtype: float64
```

In [80]:
```
PEP.loc[PEP.GarageYrBlt.isna(),'GarageYrBlt'].shape
```

Out[80]:
```
(81,)
```

In [90]:
```
PEP.GarageYrBlt.median()
```

Out[90]:
```
1980.0
```

In [91]:
```
data_imput_2 = PEP.fillna(PEP.GarageYrBlt.median())
```

In [92]:
```
data_imput_2.loc[PEP.GarageYrBlt.isna(),'GarageYrBlt']
```

Out[92]:
```
39      1980.0
48      1980.0
78      1980.0
88      1980.0
89      1980.0
         ...
1349    1980.0
1407    1980.0
1449    1980.0
1450    1980.0
1453    1980.0
Name: GarageYrBlt, Length: 81, dtype: float64
```

3(b) Identify the skewness and distribution

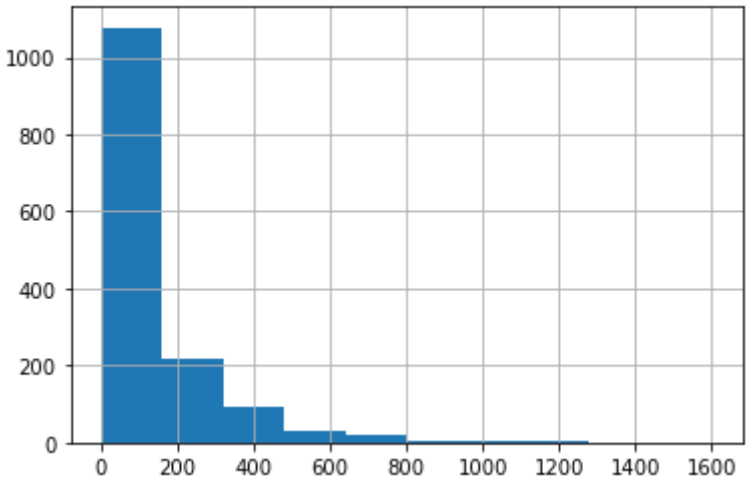
In [364…
```
# Skewness and distribution before missing value data imputation
```

In [99]:
```
PEP.MasVnrArea.skew()
```

Out[99]:
```
2.669084210182863
```

In [85]:
```
PEP.MasVnrArea.hist()
```

Out[85]:
```
<AxesSubplot:>
```



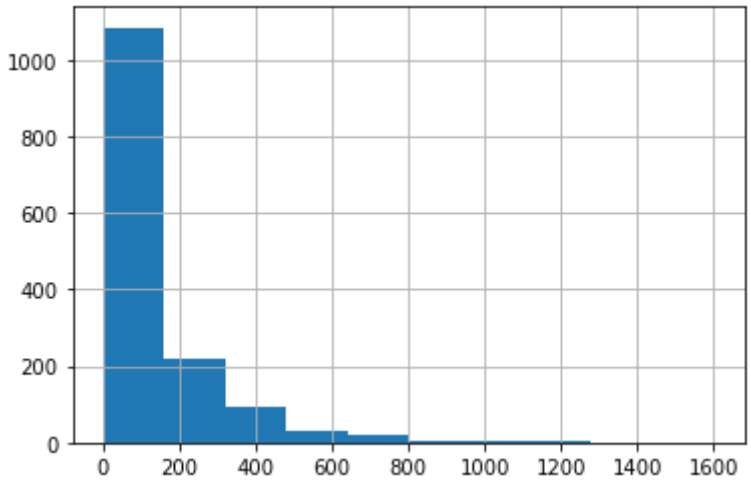
In [365…
```
# Skewness and distribution after missing value data imputation
```

In [100…
```
data_imput_1.MasVnrArea.skew()
```

Out[100]:
```
2.6776164510820997
```

In [98]:
```
data_imput_1.MasVnrArea.hist()
```

Out[98]:
```
<AxesSubplot:>
```



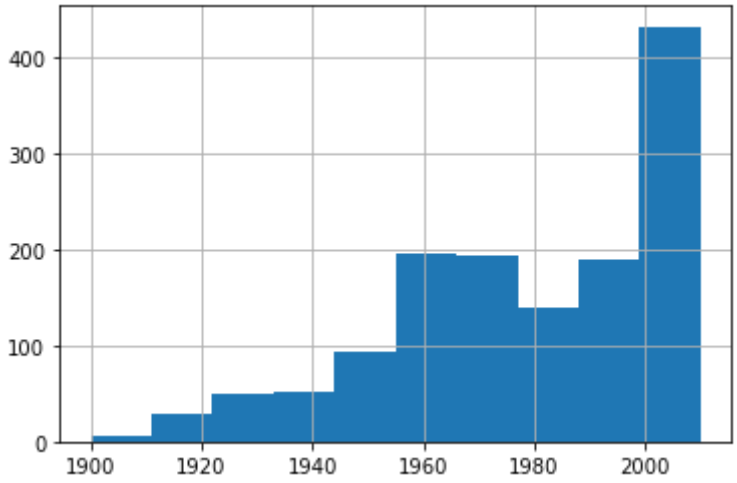
In [366…
```
# Skewness and distribution before missing value data imputation
```

In [101…
```
PEP.GarageYrBlt.skew()
```

Out[101]:
```
-0.6494146238714679
```

In [93]:
```
PEP.GarageYrBlt.hist()
```

Out[93]:
```
<AxesSubplot:>
```



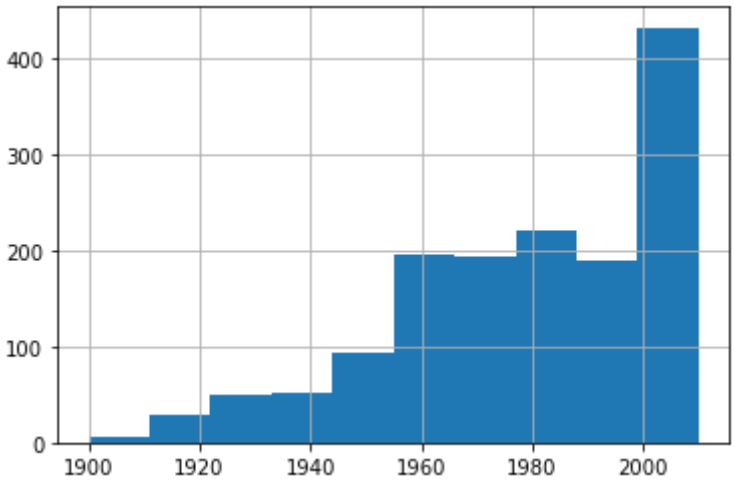
In [367…
```
# Skewness and distribution after missing value data imputation
```

In [102…
```
data_imput_2.GarageYrBlt.skew()
```

Out[102]:
```
-0.6783329490955604
```

In [94]:
```
data_imput_2.GarageYrBlt.hist()
```

Out[94]:
```
<AxesSubplot:>
```



3(C) Identify significant variables using a correlation matrix

In [103…
```
import matplotlib.pyplot as plt
```

In [104…
```
import seaborn as sns
```

In [307…
```
numeric_data.corr()
```

Out[307]:

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBu
Id	1.000000	0.011156	-0.010601	-0.033226	-0.028365	0.012609	-0.0127
MSSubClass	0.011156	1.000000	-0.386347	-0.139781	0.032628	-0.059316	0.0278
LotFrontage	-0.010601	-0.386347	1.000000	0.426095	0.251646	-0.059213	0.1233
LotArea	-0.033226	-0.139781	0.426095	1.000000	0.105806	-0.005636	0.0142
OverallQual	-0.028365	0.032628	0.251646	0.105806	1.000000	-0.091932	0.5723
OverallCond	0.012609	-0.059316	-0.059213	-0.005636	-0.091932	1.000000	-0.3759
YearBuilt	-0.012713	0.027850	0.123349	0.014228	0.572323	-0.375983	1.0000
YearRemodAdd	-0.021998	0.040581	0.088866	0.013788	0.550684	0.073741	0.5928
MasVnrArea	-0.050298	0.022936	0.193458	0.104160	0.411876	-0.128101	0.3157
BsmtFinSF1	-0.005024	-0.069836	0.233633	0.214103	0.239666	-0.046231	0.2495
BsmtFinSF2	-0.005968	-0.065649	0.049900	0.111170	-0.059119	0.040229	-0.0491
BsmtUnfSF	-0.007940	-0.140759	0.132644	-0.002618	0.308159	-0.136841	0.1490
TotalBsmtSF	-0.015415	-0.238518	0.392075	0.260833	0.537808	-0.171098	0.3914
1stFlrSF	0.010496	-0.251758	0.457181	0.299475	0.476224	-0.144203	0.2819
2ndFlrSF	0.005590	0.307886	0.080177	0.050986	0.295493	0.028942	0.0103
LowQualFinSF	-0.044230	0.046474	0.038469	0.004779	-0.030429	0.025494	-0.1837
GrLivArea	0.008273	0.074853	0.402797	0.263116	0.593007	-0.079686	0.1990
BsmtFullBath	0.002289	0.003491	0.100949	0.158155	0.111098	-0.054942	0.1875
BsmtHalfBath	-0.020155	-0.002333	-0.007234	0.048046	-0.040150	0.117821	-0.0381
FullBath	0.005587	0.131608	0.198769	0.126031	0.550600	-0.194149	0.4682
HalfBath	0.006784	0.177354	0.053532	0.014259	0.273458	-0.060769	0.2426
BedroomAbvGr	0.037719	-0.023438	0.263170	0.119690	0.101676	0.012980	-0.0706
KitchebvGr	0.002951	0.281721	-0.006069	-0.017784	-0.183882	-0.087001	-0.1748
TotRmsAbvGrd	0.027239	0.040380	0.352096	0.190015	0.427452	-0.057583	0.0955
Fireplaces	-0.019772	-0.045569	0.266639	0.271364	0.396765	-0.023820	0.1477
GarageYrBlt	0.000072	0.085072	0.070250	-0.024947	0.547766	-0.324297	0.8256
GarageCars	0.016570	-0.040110	0.285691	0.154871	0.600671	-0.185758	0.5378
GarageArea	0.017634	-0.098672	0.344997	0.180403	0.562022	-0.151521	0.4789
WoodDeckSF	-0.029643	-0.012579	0.088521	0.171698	0.238923	-0.003334	0.2248
OpenPorchSF	-0.000477	-0.006100	0.151972	0.084774	0.308819	-0.032589	0.1886
EnclosedPorch	0.002889	-0.012037	0.010700	-0.018340	-0.113937	0.070356	-0.3872
3SsnPorch	-0.046635	-0.043825	0.070029	0.020423	0.030371	0.025504	0.0313
ScreenPorch	0.001330	-0.026030	0.041383	0.043160	0.064886	0.054811	-0.0503
PoolArea	0.057044	0.008283	0.206167	0.077672	0.065166	-0.001985	0.0049
MiscVal	-0.006242	-0.007683	0.003368	0.038068	-0.031406	0.068777	-0.0343
MoSold	0.021172	-0.013585	0.011200	0.001205	0.070815	-0.003511	0.0123

	Id	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBu
YrSold	0.000712	-0.021407	0.007450	-0.014261	-0.027347	0.043950	-0.013
SalePrice	-0.021917	-0.084284	0.351799	0.263843	0.790982	-0.077856	0.522

38 rows × 38 columns

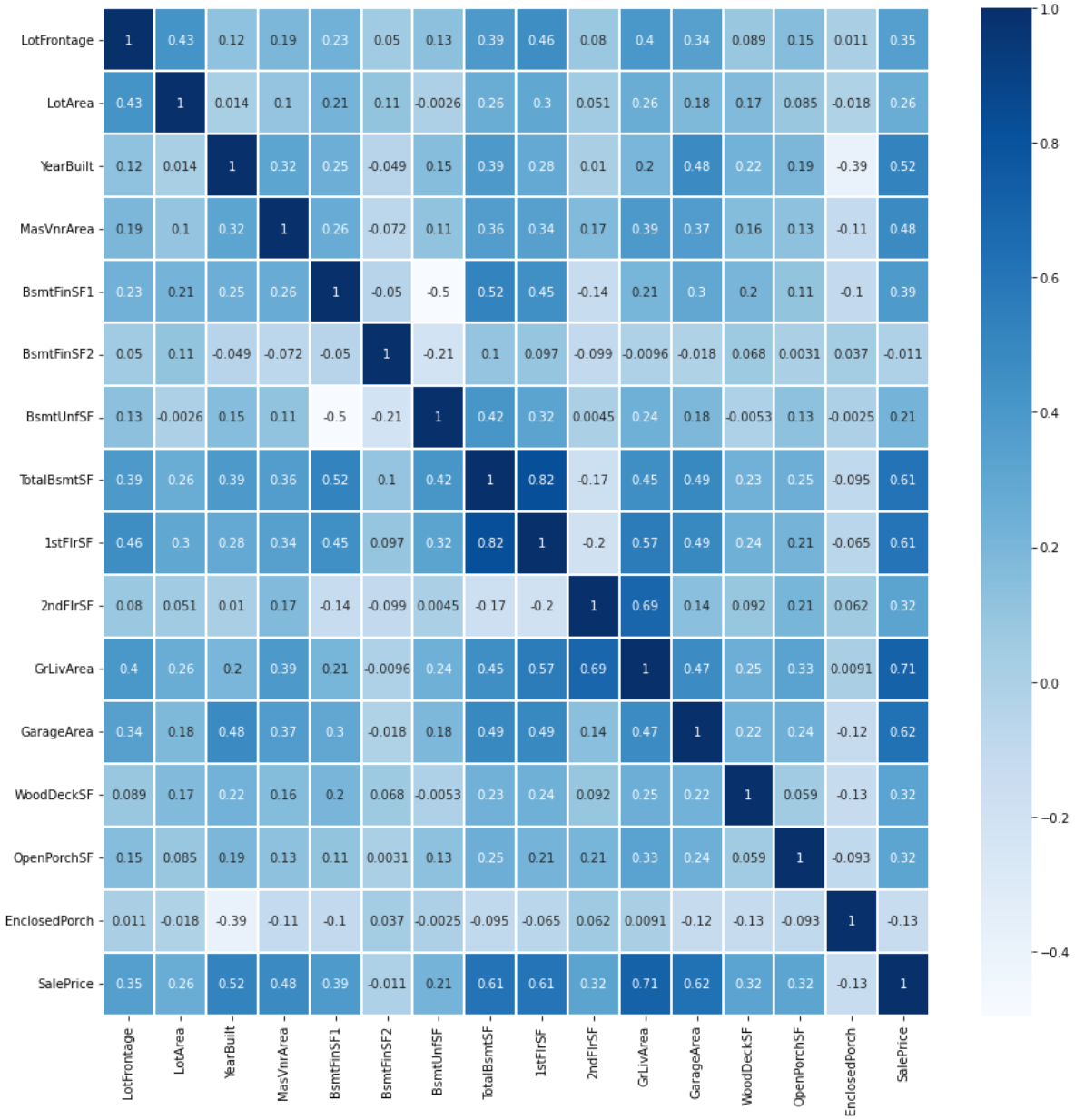
In [308…
```
for col in numeric_data.dtypes[numeric_data.dtypes != 'object'].index:

    print('No of unique values in ',col,numeric_data[col].nunique())
```

```
No of unique values in  Id 1460
No of unique values in  MSSubClass 15
No of unique values in  LotFrontage 110
No of unique values in  LotArea 1073
No of unique values in  OverallQual 10
No of unique values in  OverallCond 9
No of unique values in  YearBuilt 112
No of unique values in  YearRemodAdd 61
No of unique values in  MasVnrArea 327
No of unique values in  BsmtFinSF1 637
No of unique values in  BsmtFinSF2 144
No of unique values in  BsmtUnfSF 780
No of unique values in  TotalBsmtSF 721
No of unique values in  1stFlrSF 753
No of unique values in  2ndFlrSF 417
No of unique values in  LowQualFinSF 24
No of unique values in  GrLivArea 861
No of unique values in  BsmtFullBath 4
No of unique values in  BsmtHalfBath 3
No of unique values in  FullBath 4
No of unique values in  HalfBath 3
No of unique values in  BedroomAbvGr 8
No of unique values in  KitchebvGr 4
No of unique values in  TotRmsAbvGrd 12
No of unique values in  Fireplaces 4
No of unique values in  GarageYrBlt 97
No of unique values in  GarageCars 5
No of unique values in  GarageArea 441
No of unique values in  WoodDeckSF 274
No of unique values in  OpenPorchSF 202
No of unique values in  EnclosedPorch 120
No of unique values in  3SsnPorch 20
No of unique values in  ScreenPorch 76
No of unique values in  PoolArea 8
No of unique values in  MiscVal 21
No of unique values in  MoSold 12
No of unique values in  YrSold 5
No of unique values in  SalePrice 663
```

In [315…
```
## Since for Heatmap we want to consider only the continuos variables(having unique
# we leave the variables which we consider as discrete variables(having unique valu

plt.figure(figsize=(15,15))
sns.heatmap(numeric_data[['LotFrontage','LotArea','YearBuilt','MasVnrArea','BsmtFi
                         ,'2ndFlrSF','GrLivArea','GarageArea','WoodDeckSF','OpenP
            annot=True,cmap='Blues',linecolor='white',linewidths=2)
plt.show()
```



In [ ]:

In [ ]:

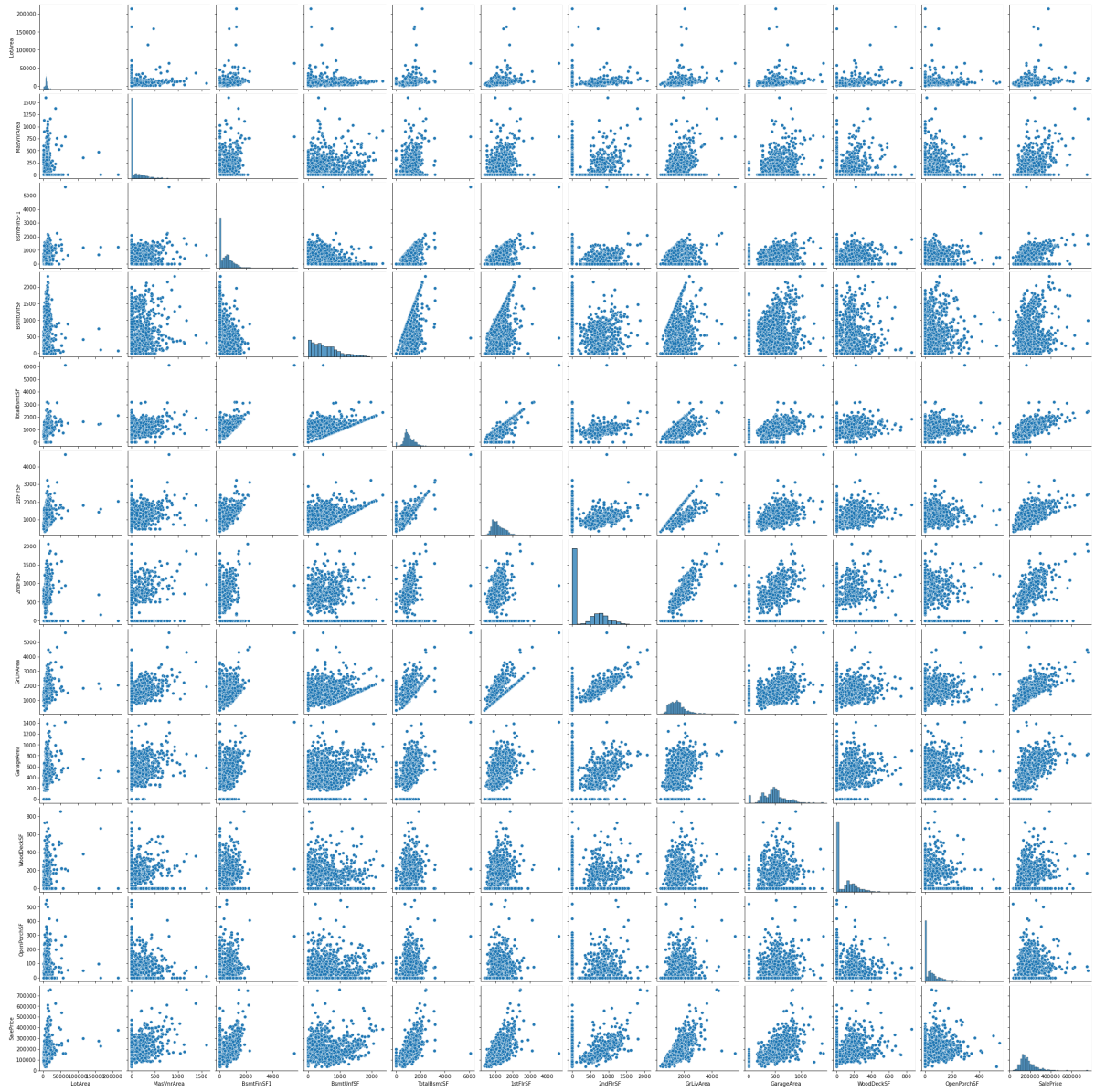
3(d) Pair plot for distribution and density

In [332…

```
## Selecting only the most continous type of variables(variables having unique valu
## for Pair plot to be more clear and understandable.

plt.figure(figsize=(7,7))
sns.pairplot(numeric_data[['LotArea','MasVnrArea','BsmtFinSF1','BsmtUnfSF','TotalBs
                           ,'2ndFlrSF','GrLivArea','GarageArea','WoodDeckSF','OpenPo
plt.show()
```

```
<Figure size 504x504 with 0 Axes>
```



In [333…
```
significant_numeric_data = (numeric_data[['LotArea','MasVnrArea','BsmtFinSF1','Bsm
                                          ,'2ndFlrSF','GrLivArea','GarageArea','WoodDeckSF','OpenP
```

In [ ]:
```

```

4) EDA of categorical variables

4(a) Missing value treatment

In [136…
```
categorical_data.isna().sum()
```

Out[136]:
```
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
Functiol           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 [138…
```
categorical_data.isna().sum()[categorical_data.isna().sum()>0]
```

Out[138]:
```
Alley          1369
MasVnrType        8
BsmtQual         37
BsmtCond         37
BsmtExposure     38
BsmtFinType1     37
BsmtFinType2     38
Electrical        1
FireplaceQu     690
GarageType       81
GarageFinish     81
GarageQual       81
GarageCond       81
PoolQC         1453
Fence          1179
MiscFeature    1406
dtype: int64
```

In [139…
```
categorical_data.isna().sum()[categorical_data.isna().sum()>0].shape
```

Out[139]:
```
(16,)
```

In [287…
```
## Percentage of missing values in each categorical variable
```

In [143…
```
percentage = (categorical_data.isna().sum()[categorical_data.isna().sum()>0]/1460)*
print(percentage)
```

```
Alley          93.767123
MasVnrType      0.547945
BsmtQual        2.534247
BsmtCond        2.534247
BsmtExposure    2.602740
BsmtFinType1    2.534247
BsmtFinType2    2.602740
Electrical      0.068493
FireplaceQu    47.260274
GarageType      5.547945
GarageFinish    5.547945
GarageQual      5.547945
GarageCond      5.547945
PoolQC         99.520548
Fence          80.753425
MiscFeature    96.301370
dtype: float64
```

In [285…
```
## Removing the categorical variables having maximum missing values
```

In [144…
```
No_miss_val = categorical_data.drop(['Alley','FireplaceQu','PoolQC','Fence','MiscF
print(No_miss_val)
```

```
     MSZoning Street LotShape LandContour Utilities LotConfig LandSlope  \
0          RL   Pave      Reg         Lvl    AllPub    Inside       Gtl
1          RL   Pave      Reg         Lvl    AllPub       FR2       Gtl
2          RL   Pave      IR1         Lvl    AllPub    Inside       Gtl
3          RL   Pave      IR1         Lvl    AllPub    Corner       Gtl
4          RL   Pave      IR1         Lvl    AllPub       FR2       Gtl
...       ...    ...      ...         ...       ...       ...       ...
1455       RL   Pave      Reg         Lvl    AllPub    Inside       Gtl
1456       RL   Pave      Reg         Lvl    AllPub    Inside       Gtl
1457       RL   Pave      Reg         Lvl    AllPub    Inside       Gtl
1458       RL   Pave      Reg         Lvl    AllPub    Inside       Gtl
1459       RL   Pave      Reg         Lvl    AllPub    Inside       Gtl

     Neighborhood Condition1 Condition2  ... Electrical KitchenQual Functiol  \
0         CollgCr       Norm       Norm  ...      SBrkr          Gd      Typ
1         Veenker      Feedr       Norm  ...      SBrkr          TA      Typ
2         CollgCr       Norm       Norm  ...      SBrkr          Gd      Typ
3         Crawfor       Norm       Norm  ...      SBrkr          Gd      Typ
4         NoRidge       Norm       Norm  ...      SBrkr          Gd      Typ
...           ...        ...        ...  ...        ...         ...      ...
1455      Gilbert       Norm       Norm  ...      SBrkr          TA      Typ
1456       NWAmes       Norm       Norm  ...      SBrkr          TA     Min1
1457      Crawfor       Norm       Norm  ...      SBrkr          Gd      Typ
1458          mes       Norm       Norm  ...      FuseA          Gd      Typ
1459      Edwards       Norm       Norm  ...      SBrkr          TA      Typ

     GarageType GarageFinish GarageQual GarageCond PavedDrive SaleType  \
0        Attchd          RFn         TA         TA          Y       WD
1        Attchd          RFn         TA         TA          Y       WD
2        Attchd          RFn         TA         TA          Y       WD
3        Detchd          Unf         TA         TA          Y       WD
4        Attchd          RFn         TA         TA          Y       WD
...         ...          ...        ...        ...        ...      ...
1455     Attchd          RFn         TA         TA          Y       WD
1456     Attchd          Unf         TA         TA          Y       WD
1457     Attchd          RFn         TA         TA          Y       WD
1458     Attchd          Unf         TA         TA          Y       WD
1459     Attchd          Fin         TA         TA          Y       WD

     SaleCondition
0           Normal
1           Normal
2           Normal
3          Abnorml
4           Normal
...            ...
1455        Normal
1456        Normal
1457        Normal
1458        Normal
1459        Normal

[1460 rows x 38 columns]
```

In [286…

```
## Percentage of missing values in each categorical variable after Removing the cat
```

In [148…

```
after_drop = No_miss_val.isna().sum()[No_miss_val.isna().sum()>0]/1460*100
print(after_drop)
```

```
MasVnrType      0.547945
BsmtQual        2.534247
BsmtCond        2.534247
BsmtExposure    2.602740
BsmtFinType1    2.534247
BsmtFinType2    2.602740
Electrical      0.068493
GarageType      5.547945
GarageFinish    5.547945
GarageQual      5.547945
GarageCond      5.547945
dtype: float64
```

In [149…
```
after_drop.shape
```

Out[149]:
```
(11,)
```

In [288…
```
## Missing Value imputation
```

In [289…
```
## (1)
```

In [237…
```
categorical_data['MasVnrType'].mode()
```

Out[237]:
```
0    None
Name: MasVnrType, dtype: object
```

In [239…
```
categorical_data['MasVnrType'].fillna('None', inplace = True)
```

In [241…
```
categorical_data['MasVnrType'].isna().sum()
```

Out[241]:
```
0
```

In [290…
```
## (2)
```

In [243…
```
categorical_data['BsmtQual'].isna().sum()
```

Out[243]:
```
37
```

In [244…
```
categorical_data['BsmtQual'].mode()
```

Out[244]:
```
0    TA
Name: BsmtQual, dtype: object
```

In [245…
```
categorical_data['BsmtQual'].fillna('None', inplace = True)
```

In [246…
```
categorical_data['BsmtQual'].isna().sum()
```

Out[246]:
```
0
```

In [291…
```
## (3)
```

In [248…
```
categorical_data['BsmtCond'].isna().sum()
```

Out[248]:
```
37
```

In [249…
```
categorical_data['BsmtCond'].mode()
```

Out[249]:
```
0    TA
Name: BsmtCond, dtype: object
```

In [250…
```
categorical_data['BsmtCond'].fillna('None', inplace = True)
```

In [251…
```
categorical_data['BsmtCond'].isna().sum()
```

Out[251]:
```
0
```

In [292…
```
## (4)
```

In [252…
```
categorical_data['BsmtExposure'].isna().sum()
```

Out[252]:
```
38
```

In [253…
```
categorical_data['BsmtExposure'].mode()
```

Out[253]:
```
0    No
Name: BsmtExposure, dtype: object
```

In [254…
```
categorical_data['BsmtExposure'].fillna('None', inplace = True)
```

In [255…
```
categorical_data['BsmtExposure'].isna().sum()
```

Out[255]:
```
0
```

In [293…
```
## (5)
```

In [256…
```
categorical_data['BsmtFinType1'].isna().sum()
```

Out[256]:
```
37
```

In [258…
```
categorical_data['BsmtFinType1'].mode()
```

Out[258]:
```
0    Unf
Name: BsmtFinType1, dtype: object
```

In [259…
```
categorical_data['BsmtFinType1'].fillna('None', inplace = True)
```

In [260…
```
categorical_data['BsmtFinType1'].isna().sum()
```

Out[260]:
```
0
```

In [294…
```
## (6)
```

In [261…
```
categorical_data['BsmtFinType2'].isna().sum()
```

Out[261]:
```
38
```

In [262…
```
categorical_data['BsmtFinType2'].mode()
```

Out[262]:
```
0    Unf
Name: BsmtFinType2, dtype: object
```

In [263…
```
categorical_data['BsmtFinType2'].fillna('None', inplace = True)
```

In [264…
```
categorical_data['BsmtFinType2'].isna().sum()
```

Out[264]:
```
0
```

In [295…
```
## (7)
```

In [265…
```
categorical_data['Electrical'].isna().sum()
```
Out[265]:
```
1
```

In [266…
```
categorical_data['Electrical'].mode()
```
Out[266]:
```
0    SBrkr
Name: Electrical, dtype: object
```

In [267…
```
categorical_data['Electrical'].fillna('None', inplace = True)
```

In [268…
```
categorical_data['Electrical'].isna().sum()
```
Out[268]:
```
0
```

In [296…
```
## (8)
```

In [269…
```
categorical_data['GarageType'].isna().sum()
```
Out[269]:
```
81
```

In [270…
```
categorical_data['GarageType'].mode()
```
Out[270]:
```
0    Attchd
Name: GarageType, dtype: object
```

In [271…
```
categorical_data['GarageType'].fillna('None', inplace = True)
```

In [272…
```
categorical_data['GarageType'].isna().sum()
```
Out[272]:
```
0
```

In [297…
```
## (9)
```

In [273…
```
categorical_data['GarageFinish'].isna().sum()
```
Out[273]:
```
81
```

In [274…
```
categorical_data['GarageFinish'].mode()
```
Out[274]:
```
0    Unf
Name: GarageFinish, dtype: object
```

In [275…
```
categorical_data['GarageFinish'].fillna('None', inplace = True)
```

In [276…
```
categorical_data['GarageFinish'].isna().sum()
```
Out[276]:
```
0
```

In [298…
```
## (10)
```

In [277…
```
categorical_data['GarageQual'].isna().sum()
```
Out[277]:
```
81
```

In [278…
```
categorical_data['GarageQual'].mode()
```
Out[278]:
```
0    TA
Name: GarageQual, dtype: object
```

In [279…
```
categorical_data['GarageQual'].fillna('None', inplace = True)
```

In [280…
```
categorical_data['GarageQual'].isna().sum()
```

Out[280]:
```
0
```

In [299…
```
## (11)
```

In [281…
```
categorical_data['GarageCond'].isna().sum()
```

Out[281]:
```
81
```

In [282…
```
categorical_data['GarageCond'].mode()
```

Out[282]:
```
0    TA
Name: GarageCond, dtype: object
```

In [283…
```
categorical_data['GarageCond'].fillna('None', inplace = True)
```

In [284…
```
categorical_data['GarageCond'].isna().sum()
```

Out[284]:
```
0
```

In [329…
```
# 'significant categorical variables' after dropping the maximum missing value var

significant_categoric_data = categorical_data.drop(['Alley','FireplaceQu','PoolQC'
```

In [330…
```
significant_categoric_data
```

Out[330]:

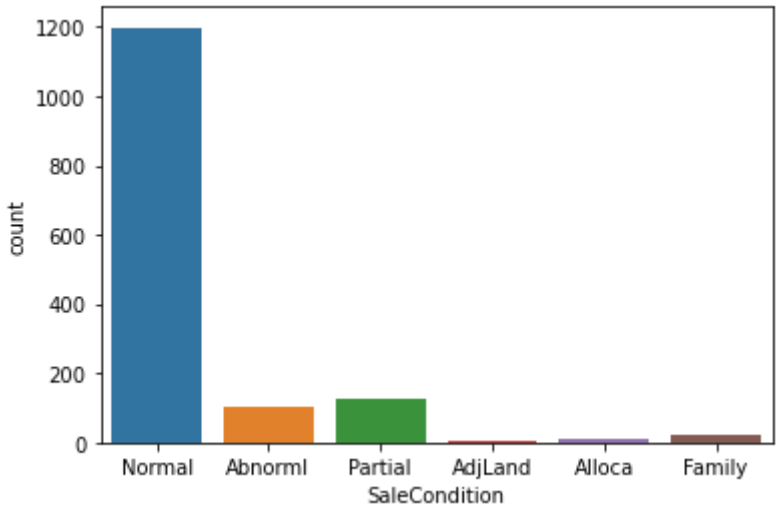
	MSZoning	Street	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood
0	RL	Pave	Reg	Lvl	AllPub	Inside	Gtl	CollgCr
1	RL	Pave	Reg	Lvl	AllPub	FR2	Gtl	Veenker
2	RL	Pave	IR1	Lvl	AllPub	Inside	Gtl	CollgCr
3	RL	Pave	IR1	Lvl	AllPub	Corner	Gtl	Crawfor
4	RL	Pave	IR1	Lvl	AllPub	FR2	Gtl	NoRidge
...	...	...	...	...	...	...	...	...
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

1460 rows × 38 columns

4(b) Count plot and box plot for bivariate analysis

In [318…
```
sns.countplot('SaleCondition',data=categorical_data)
```

Out[318]:
```
<AxesSubplot:xlabel='SaleCondition', ylabel='count'>
```

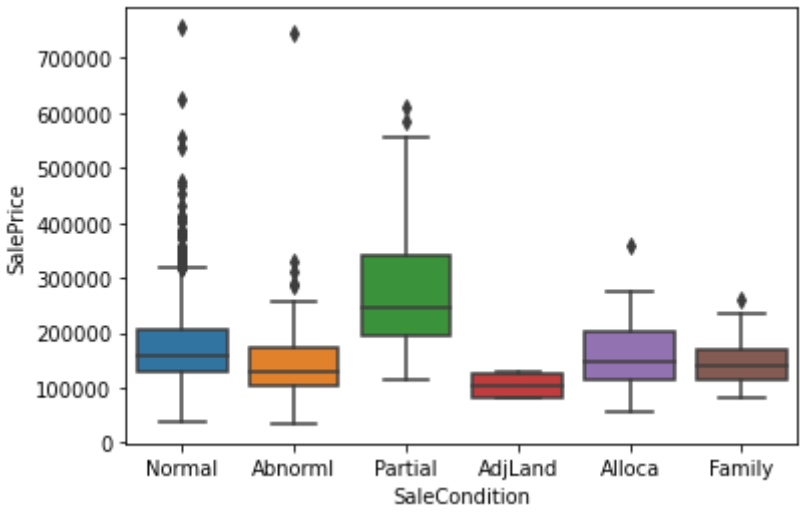


In [321…
```
## Box plot for bivariate analysis.

sns.boxplot('SaleCondition','SalePrice', data = PEP)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarnin
g: Pass the following variables as keyword args: x, y. From version 0.12, the only
valid positional argument will be `data`, and passing other arguments without an e
xplicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[321]:
```
<AxesSubplot:xlabel='SaleCondition', ylabel='SalePrice'>
```



5) Combine all the significant categorical and numerical variables

In [331…
```
significant_categoric_data
```

Out[331]:

	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

1460 rows × 38 columns

In [336...
```
significant_numeric_data
```

Out[336]:

	LotArea	MasVnrArea	BsmtFinSF1	BsmtUnfSF	TotalBsmtSF	1stFlrSF	2ndFlrSF	GrLivArea
0	8450	196.0	706	150	856	856	854	1710
1	9600	0.0	978	284	1262	1262	0	1262
2	11250	162.0	486	434	920	920	866	1786
3	9550	0.0	216	540	756	961	756	1717
4	14260	350.0	655	490	1145	1145	1053	2198
...	...	...	...	...	...	...	...	...
1455	7917	0.0	0	953	953	953	694	1647
1456	13175	119.0	790	589	1542	2073	0	2073
1457	9042	0.0	275	877	1152	1188	1152	2340
1458	9717	0.0	49	0	1078	1078	0	1078
1459	9937	0.0	830	136	1256	1256	0	1256

1460 rows × 12 columns

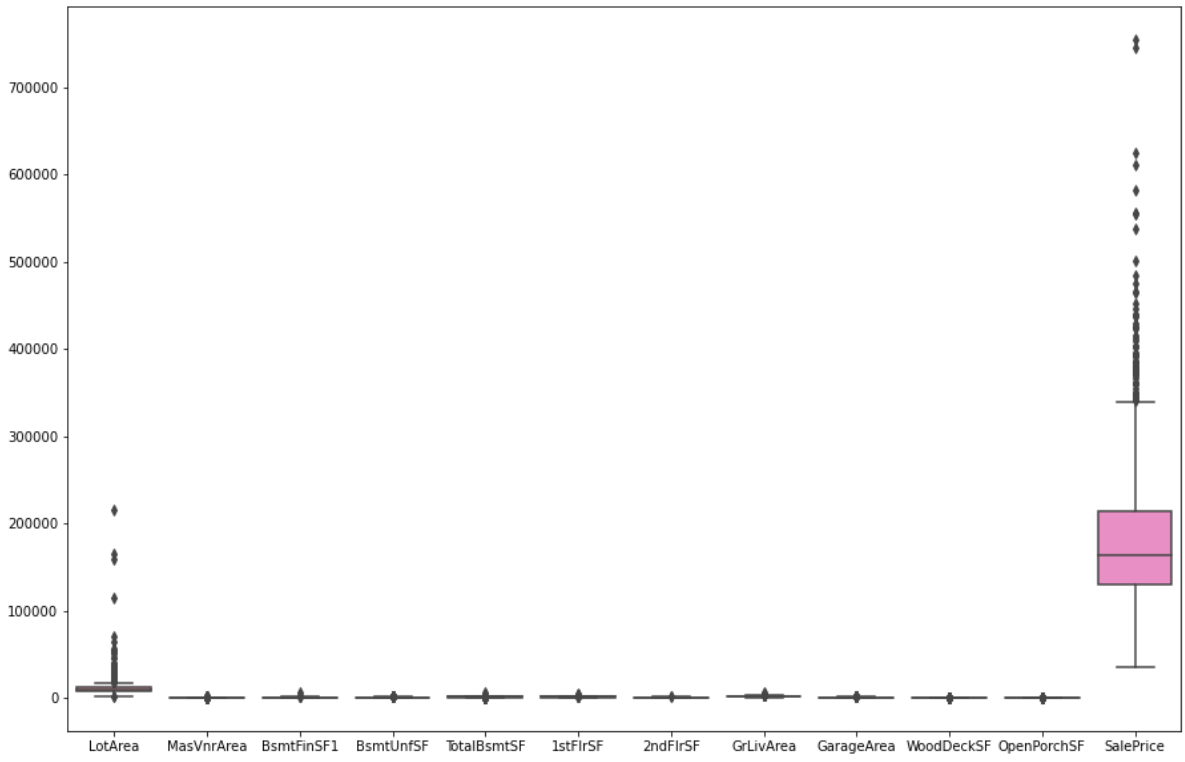
In [337...
```
df1 = significant_categoric_data
```

In [338...
```
df2 = significant_numeric_data
```

6)Plot box plot for the new dataset to find the variables with outliers

In [350…
```
plt.figure(figsize=(15,10))
sns.boxplot( data = df2)
```

Out[350]:
```
<AxesSubplot:>
```



In [ ]:

In [ ]:

In [ ]: