```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn as sk
%matplotlib inline
from sklearn.model_selection import train_test_split
```

Understand the dataset:

1a. Identify the shape of the dataset

```
train_data = pd.read_csv('train.csv')
train_data.shape
```

```
(1460, 81)
```

The PDF dataset has 1460 rows and 80 columns

1b. Identify variables with unique values

```
train_data.nunique()
```

```
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
Exterior1       15
Exterior2       16
MasVnrType       4
MasVnrArea     327
ExterQual        4
ExterCond        4
Foundation       6
BsmtQual         4
BsmtCond         4
BsmtExposure     4
BsmtFinType1     6
BsmtFinSF1     637
BsmtFinType2     6
BsmtFinSF2     144
BsmtUnfSF      780
TotalBsmtSF    721
Heating          6
HeatingQC        5
CentralAir       2
Electrical       5
1stFlrSF       753
2ndFlrSF       417
LowQualFinSF    24
GrLivArea      861
BsmtFullBath     4
BsmtHalfBath     3
FullBath         4
HalfBath         3
BedroomAbvGr     8
KitchenAbvGr     4
KitchenQual      4
TotRmsAbvGrd    12
Functional       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       24
ScreenPorch     92
PoolArea        13
PoolQC           3
Fence            4
MiscFeature      4
MiscVal         21
MoSold          12
YrSold           5
SalePrice      663
dtype: int64
```

1b. Identify variables with unique values

```
train_data.unique()
```

```
MSSubClass      [60, 20, 70, 50, 190, 45, 90, 120, 30, 80, 1...
MSZoning        [RL, RM, C (all), FV, RH]
LotFrontage     [65.0, 80.0, 68.0, 60.0, 84.0, 85.0, 75.0, n...
LotArea         [8450, 9600, 11250, 10084, 10382, 6120, 7420...
Street          [Pave, Grvl]
Alley           [nan, Grvl, Pave]
LotShape        [Reg, IR1, IR2, IR3]
LandContour     [Lvl, Bnk, HLS, Low]
Utilities       [AllPub, NoSeWa]
LotConfig       [Inside, FR2, Corner, CulDSac, FR3]
LandSlope       [Gtl, Mod, Sev]
Neighborhood    [CollgCr, Veenker, Crawfor, NoRidge, Mitchel...
Condition1      [Norm, Feedr, PosN, Artery, RRAe, RRNn, RRAn...
Condition2      [Norm, RRNn, Feedr, PosN, Artery, RRAe]
BldgType        [1Fam, 2fmCon, Duplex, TwnhsE, Twnhs]
HouseStyle      [2Story, 1Story, 1.5Fin, 1.5Unf, SLvl, SFoye...
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, 1993, 1004, 1973, 1...
YearRemodAdd    [2003, 1976, 2002, 1970, 1995, 1950, 1965, 1...
RoofStyle       [Gable, Hip, Gambrel, Mansard, Flat, Shed]
RoofMatl        [CompShg, WdShake, Metal, WdShngl, Membran, ...
Exterior1       [VinylSd, MetalSd, Wd Sdng, HdBoard, BrkFace...
Exterior2       [VinylSd, MetalSd, Wd Shng, HdBoard, Plywood...
MasVnrType      [BrkFace, None, Stone, BrkCmn]
MasVnrArea      [196.0, 0.0, 162.0, 350.0, 186.0, 240.0, 286...
ExterQual       [Gd, TA, Ex, Fa]
ExterCond       [TA, Gd, Fa, Po, Ex]
Foundation      [PConc, CBlock, BrkTil, Slab, Stone, Wood]
BsmtQual        [Gd, TA, Ex, nan]
BsmtCond        [TA, Gd, nan, Fa, Po]
BsmtExposure    [No, Gd, Mn, Av, nan]
BsmtFinType1    [GLQ, ALQ, Unf, Rec, BLQ, nan]
BsmtFinSF1      [706, 978, 486, 216, 655, 732, 1369, 859, 771...
BsmtFinType2    [Unf, BLQ, nan, ALQ, Rec, LwQ]
BsmtFinSF2      [0, 354, 732, 123, 146, 1120, 374, 393, 363,...
BsmtUnfSF       [150, 284, 434, 540, 490, 64, 317, 216, 228,...
TotalBsmtSF     [856, 1262, 920, 961, 1145, 796, 1686, 1107,...
Heating         [GasA, GasW, Grav, Wall, OthW, Floor]
HeatingQC       [Ex, Gd, TA, Fa, Po]
CentralAir      [Y, N]
Electrical      [SBrkr, FuseF, FuseA, FuseP, Mix]
1stFlrSF        [856, 1262, 920, 961, 1145, 796, 1694, 1107,...
2ndFlrSF        [854, 0, 866, 756, 1053, 566, 983, 642, 907,...
LowQualFinSF    [0, 360, 513, 234, 528, 572, 144, 392, 371, ...
GrLivArea       [1710, 1262, 1786, 1717, 2198, 1362, 1694, 2...
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]
Functional      [Typ, Min1, Maj1, Min2, Mod, Maj2, Sev]
Fireplaces      [0, 1, 2, 3]
FireplaceQu     [nan, TA, Gd, Fa, Ex]
GarageType      [Attchd, Detchd, BuiltIn, CarPort, nan, Basment]
GarageYrBlt     [2003, 1976, 1977, 1999, 1931, 1884, 1908, 19...
GarageFinish    [RFn, Unf, Fin]
GarageCars      [2, 3, 1, 0, 4]
GarageArea      [548, 460, 608, 642, 836, 480, 636, 484, 468,...
GarageQual      [TA, Fa, Gd, nan, Ex, Po]
GarageCond      [TA, Fa, nan, Gd, Po, Ex]
PavedDrive      [Y, N, P]
WoodDeckSF      [0, 298, 192, 40, 255, 235, 90, 147, 241, 362...
OpenPorchSF     [61, 0, 42, 307, 84, 128, 158, 99, 169, 210, ...
EnclosedPorch   [0, 112, 108, 272, 228, 205, 172, 144, 96, 120, ...
3SsnPorch       [0, 320, 168, 304, 180, 236, 162, 270, 260, 1...
ScreenPorch     [0, 176, 192, 228, 205, 90, 264, 80, 224, 100...
PoolArea        [0, 512, 648, 576, 555, 480, 519]
PoolQC          [nan, Ex, Fa, Gd]
Fence           [nan, MnPrv, GdWo, GdPrv, MnWw]
MiscFeature     [nan, Shed, Othr, Gar2, TenC]
MiscVal         [0, 400, 500, 700, 350, 2000, 600, 54, 1200, ...
MoSold          [2, 5, 9, 12, 8, 11, 7, 1, 10, 4, 6, 3]
YrSold          [2008, 2007, 2006, 2009, 2010]
SalePrice       [208500, 181500, 223500, 140000, 250000, 143...
dtype: object
```

All variables have unique values

2. Generate a separate dataset for numerical and categorical variables

```
numerical_train_df=train_data._get_numeric_data()
categorical_train_df=train_data[train_data.select_dtypes(exclude=['int64','float64'])]
```

```
numerical_train_df
```

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	Bs
0	60	65.0	8450	7	5	2003	2003	196.0	706	0	
1	20	80.0	9600	6	8	1976	1976	0.0	978	0	
2	70	68.0	11250	7	5	2001	2002	162.0	486	0	
3	50	60.0	10084	5	8	1915	1970	0.0	216	0	
4	190	84.0	10382	5	5	1993	1995	350.0	655	0	
...	...	...	...	...	...	...	...	...	...	...	
1455	60	62.0	7917	6	5	1999	2000	0.0	0	0	
1456	20	85.0	13175	6	6	1978	1988	119.0	790	163	
1457	70	66.0	9042	7	9	1941	2006	0.0	275	0	
1458	20	68.0	9717	5	6	1950	1996	0.0	49	1029	
1459	20	75.0	9937	5	6	1965	1965	0.0	830	290	

1460 rows × 37 columns

numerical data has 1460 rows and 37 columns

3 EDA of Numerical Variables

3a. Missing value treatment

```
numerical_train_df.isnull().sum(axis=0).sort_values(ascending=False)
```

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

```
numerical_train_df[numerical_train_df.select_dtypes(exclude=['int64','float64'])].isnull().sum(axis=0)
```

```
Series([], dtype: int64)
```

There are 3 variables (LotFrontage, MasVnrArea, GarageYrBlt) in numeric data that have null values.

```
numerical_train_df.hist(figsize=(20,20))
```

The distribution plot of 'LotFrontage' show nearly normal distribution and 'GarageYrBlt' show negatively skewed distribution.

```
numerical_train_df[numerical_train_df.select_dtypes(exclude=['int64','float64'])].hist(figsize=(20,20))
```

```
array([[<AxesSubplot:>]], dtype=object)
```

GarageYrBlt column has float values. Therefore, Ismal asked me to drop the column.

```
numerical_train_df.isnull.sum(axis=0).sort_values(ascending=False)
```

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	Bs
0	60	65.0	8450	7	5	2003	2003	196.0	706	0	
1	20	80.0	9600	6	8	1976	1976	0.0	978	0	
2	70	68.0	11250	7	5	2001	2002	162.0	486	0	
3	50	60.0	10084	5	8	1915	1970	0.0	216	0	
4	190	84.0	10382	5	5	1993	1995	350.0	655	0	
...	...	...	...	...	...	...	...	...	...	...	
1455	60	62.0	7917	6	5	1999	2000	0.0	0	0	
1456	20	85.0	13175	6	6	1978	1988	119.0	790	163	
1457	70	66.0	9042	7	9	1941	2006	0.0	275	0	
1458	20	68.0	9717	5	6	1950	1996	0.0	49	1029	
1459	20	75.0	9937	5	6	1965	1965	0.0	830	290	

1460 rows × 37 columns

LotFrontage null values are treated using median of the column.

GarageYrBlt is deleted as it is showing float values.

MasVnrArea null values are replaced with zero as MasVnrType is also null.

```
numerical_train_df.describe(percentile=[.1,.2,.3,.4,.5,.6,.7,.8,.9,.99,.100])
```

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	Bs
count	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000
mean	56.897260	69.863699	10516.828082	6.099315	5.575342	1971.267808	1984.865753	103.685262	443.639726	46.549315	567.240411
std	42.300571	22.026051	9981.264932	1.382997	1.112799	30.202904	20.645407	181.066207	456.098091	161.319273	441.369461
min	20.000000	21.000000	1300.000000	1.000000	1.000000	1872.000000	1950.000000	0.000000	0.000000	0.000000	0.000000
10%	20.000000	50.000000	5000.000000	4.000000	4.000000	1900.000000	1950.000000	0.000000	0.000000	0.000000	0.000000
20%	20.000000	60.000000	6000.000000	5.000000	5.000000	1930.000000	1960.000000	0.000000	0.000000	0.000000	0.000000
30%	20.000000	65.000000	7000.000000	5.000000	5.000000	1950.000000	1970.000000	0.000000	0.000000	0.000000	0.000000
40%	20.000000	69.000000	7879.000000	5.800000	5.000000	1962.000000	1980.000000	0.000000	350.000000	0.000000	465.000000
50%	50.000000	70.000000	9478.500000	6.000000	5.000000	1971.000000	1993.000000	0.000000	477.500000	0.000000	559.500000
60%	60.000000	72.000000	10800.000000	6.000000	6.000000	1978.000000	2000.000000	108.000000	570.000000	0.000000	672.000000
70%	70.000000	75.000000	11500.000000	7.000000	6.000000	1984.000000	2004.000000	216.000000	662.000000	0.000000	772.000000
80%	80.000000	78.000000	12546.000000	7.000000	7.000000	1992.000000	2006.000000	340.000000	821.000000	0.000000	882.000000
90%	160.000000	82.000000	14381.000000	8.000000	8.000000	2000.000000	2007.000000	573.000000	1065.000000	350.000000	1232.000000
99%	190.000000	107.000000	21930.000000	9.000000	9.000000	2007.000000	2009.000000	1047.000000	1462.000000	1474.000000	2153.000000
100%	190.000000	313.000000	215245.000000	10.000000	9.000000	2010.000000	2010.000000	1600.000000	5644.000000	1474.000000	6110.000000

8 rows × 36 columns

```
numerical_train_df.skew(axis=0)
```

```
MSSubClass       1.407657
LotFrontage      2.163569
LotArea         12.207688
OverallQual      0.216944
OverallCond      0.692871
YearBuilt       -0.613461
YearRemodAdd    -0.503562
MasVnrArea       2.669084
BsmtFinSF1       1.683771
BsmtFinSF2       4.255261
BsmtUnfSF        0.919689
TotalBsmtSF      1.524255
1stFlrSF         1.376757
2ndFlrSF         0.813030
LowQualFinSF     9.011171
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
GarageCars      -0.342549
GarageArea       0.179981
WoodDeckSF       1.541376
OpenPorchSF      2.364342
EnclosedPorch    4.003891
3SsnPorch       10.304342
ScreenPorch      4.122214
PoolArea        14.828374
MiscVal         24.476794
MoSold           0.212053
YrSold           0.096269
SalePrice        1.882876
dtype: 0.096269
```

The above list shows variables that have skewed values.

```
numerical_train_df.kurt()
```

Inference: The skewness value of the SalePrice depicts that the SalePrice dataset is not normally distributed. Therefore, we need to normalize it.

3c. Identify significant variables using a correlation matrix

```
numerical_train_df.corr(axis=0)
```

	MSSubClass	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinSF2	Bs
MSSubClass	1.000000	-0.386347	-0.139781	0.032628	-0.059316	0.027850	0.040581	0.022936	-0.069836	-0.065718	-0.140759
LotFrontage	-0.386347	1.000000	0.426095	0.251646	-0.059213	0.123349	0.088866	0.193458	0.233633	0.049900	0.132644
LotArea	-0.139781	0.426095	1.000000	0.105806	-0.005636	0.014228	0.013788	0.104160	0.214103	0.111170	0.002618
OverallQual	0.032628	0.251646	0.105806	1.000000	-0.091932	0.593007	0.550684	0.411876	0.239666	-0.059119	0.308159
OverallCond	-0.059316	-0.059213	-0.005636	-0.091932	1.000000	-0.375983	0.073741	-0.128101	-0.046231	0.040229	-0.136841
YearBuilt	0.027850	0.123349	0.014228	0.593007	-0.375983	1.000000	0.592855	0.312018	0.249503	-0.049107	0.149040
YearRemodAdd	0.040581	0.088866	0.013788	0.550684	0.073741	0.592855	1.000000	0.175337	0.128451	-0.067759	0.181133
MasVnrArea	0.022936	0.193458	0.104160	0.411876	-0.128101	0.312018	0.175337	1.000000	0.261256	-0.072318	0.113036
BsmtFinSF1	-0.069836	0.233633	0.214103	0.239666	-0.046231	0.249503	0.128451	0.261256	1.000000	-0.050117	0.558622
BsmtFinSF2	-0.065718	0.049900	0.111170	-0.059119	0.040229	-0.049107	-0.067759	-0.072318	-0.050117	1.000000	-0.088871
BsmtUnfSF	-0.140759	0.132644	0.002618	0.308159	-0.136841	0.149040	0.181133	0.113036	0.558622	-0.088871	1.000000
TotalBsmtSF	-0.238518	0.392075	0.260833	0.537808	-0.171098	0.391452	0.291066	0.363936	0.522396	0.104810	0.415360
1stFlrSF	-0.251758	0.457181	0.299475	0.476224	-0.144203	0.281986	0.240379	0.344501	0.445863	0.097117	0.318075
2ndFlrSF	0.307886	0.080177	0.050986	0.295493	0.028942	0.010308	0.140024	0.174256	-0.137079	-0.099260	-0.002538
LowQualFinSF	-0.372354	0.037041	0.004779	-0.030429	0.025494	-0.183784	-0.062419	-0.069480	-0.064503	0.014807	0.037963
GrLivArea	0.074034	0.402797	0.263116	0.593557	-0.079686	0.199010	0.287389	0.390857	0.208171	0.009314	0.220192
BsmtFullBath	-0.023047	0.037737	0.158155	0.111098	-0.054942	0.187599	0.119470	0.083848	0.649212	0.158678	0.120981
BsmtHalfBath	-0.008555	-0.005030	0.054851	-0.040150	0.117821	-0.038162	-0.038555	0.011128	0.067418	0.070488	-0.025844
FullBath	0.131608	0.198769	0.126031	0.550600	-0.194149	0.468271	0.439046	0.288486	0.058543	-0.076444	0.035353
HalfBath	0.177354	0.053532	0.014259	0.273458	-0.060769	0.242656	0.183331	0.040380	0.004262	-0.032148	0.044413
BedroomAbvGr	0.251732	0.263170	0.119690	0.101676	0.012980	-0.070651	-0.045286	0.102821	0.049900	-0.022736	0.141290
KitchenAbvGr	0.281721	-0.006069	-0.018772	-0.183882	-0.087001	-0.174800	-0.149598	-0.037610	-0.096683	-0.053672	0.137733
TotRmsAbvGrd	0.732821	0.352096	0.190015	0.427452	-0.057583	0.095589	0.191740	0.280682	0.044316	-0.035227	0.291066
Fireplaces	0.046474	0.266639	0.271364	0.396671	-0.023820	0.147716	0.112581	0.247033	0.260011	0.046921	0.327582
GarageYrBlt	0.074853	0.070250	0.053984	0.547766	-0.324297	0.825667	0.642277	0.252691	0.153276	-0.037963	0.150673
GarageCars	-0.040110	0.285691	0.154871	0.600671	-0.185758	0.537850	0.420622	0.364204	0.224666	-0.038264	0.214070
GarageArea	-0.098672	0.344997	0.180403	0.562022	-0.151521	0.478954	0.371600	0.373066	0.296987	-0.018227	0.182511
WoodDeckSF	-0.012037	0.088521	0.171698	0.238923	-0.003334	0.224880	0.205726	0.159728	0.204306	0.067898	0.096294
OpenPorchSF	-0.043825	0.151972	0.084774	0.308819	-0.032589	0.188686	0.226298	0.125703	0.111170	0.003093	0.116960
EnclosedPorch	0.084848	-0.010672	-0.018227	-0.113037	0.070356	-0.387268	-0.193919	-0.099555	-0.071188	0.036543	0.060762
3SsnPorch	0.028358	0.070029	0.043160	0.030371	0.025504	0.031037	0.045286	0.018796	0.054601	-0.029993	0.044584
ScreenPorch	0.000354	0.041383	0.077672	0.064886	0.054811	0.070231	0.062157	0.027081	0.043772	0.000704	0.074181
PoolArea	0.057044	0.206223	0.038068	0.065166	-0.001985	0.004950	0.044584	0.064774	0.075989	0.060021	0.100772
MiscVal	-0.006952	0.003368	-0.031408	-0.027837	0.044689	-0.034383	-0.050364	-0.011378	-0.006434	0.004940	-0.036420
MoSold	0.022432	0.011200	0.001884	0.070815	-0.058771	0.012398	0.021676	0.022420	0.037738	0.005723	0.034888
YrSold	-0.027168	0.007450	-0.014314	-0.027571	0.043746	-0.013618	0.035743	-0.048918	-0.005024	0.037099	-0.020454
SalePrice	-0.084284	0.351799	0.263843	0.790982	-0.077856	0.522897	0.507101	0.472614	0.386420	-0.011378	0.214479

38 rows × 38 columns

```
plt.figure(figsize=(40,40))
sns.heatmap(numerical_train_df.corr(),annot=True,cmap='Blues')
```

```
numerical_train_df.corr(axis=0).sort_values(ascending=False)
```

```
SalePrice      1.000000
OverallQual    0.790982
GrLivArea      0.708624
GarageCars     0.640401
GarageArea     0.623431
TotalBsmtSF    0.613581
1stFlrSF       0.605852
FullBath       0.560664
TotRmsAbvGrd   0.533723
YearBuilt      0.522897
YearRemodAdd   0.507101
GarageYrBlt    0.486362
MasVnrArea     0.472614
Fireplaces     0.466929
BsmtFinSF1     0.386420
WoodDeckSF     0.324413
2ndFlrSF       0.319334
OpenPorchSF    0.315856
HalfBath       0.284108
LotArea        0.263843
BsmtFullBath   0.227122
BsmtUnfSF      0.214479
BedroomAbvGr   0.168213
screenArea     0.122508
PoolArea       0.092404
MoSold         0.046432
3SsnPorch      0.044584
BsmtFinSF2     0.005148
BsmtHalfBath   0.016844
MiscVal        0.021190
LowQualFinSF   0.037963
YrSold         0.028923
OverallCond    0.077856
MSSubClass     0.084284
EnclosedPorch  0.128578
KitchenAbvGr   0.135907
LotFrontage    0.351799
ScreenPorch    0.111447
dtype: 0.111447
```

```
numerical_train_df[numerical_train_df.select_dtypes(exclude=['int64','float64'])].corr(axis=0)
```

```
Series([], dtype: int64)
```

with threshold ±0.5 or ±0.5, the above listed numerical data variables are the ones that have high correlation with SalePrice

3d. Pair plot for distribution and density

```
sns.pairplot(numerical_train_df,vars=['OverallQual','GrLivArea','GarageCars','GarageArea','TotalBsmtSF','1stFlrSF','FullBath','TotRmsAbvGrd','YearBuilt','YearRemodAdd','GarageYrBlt','MasVnrArea','Fireplaces','BsmtFinSF1','WoodDeckSF','2ndFlrSF','OpenPorchSF','HalfBath','LotArea','BsmtFullBath','BsmtUnfSF','BedroomAbvGr','screenArea','PoolArea','MoSold','3SsnPorch','BsmtFinSF2','BsmtHalfBath','MiscVal','LowQualFinSF','YrSold','OverallCond','MSSubClass','EnclosedPorch','KitchenAbvGr','LotFrontage','ScreenPorch','SalePrice'])
```

OverallQual is directly related with SalePrice. As the OverallQual is going higher upto 10, there is increase in SalePrice.

TotalBsmtSF is directly related with SalePrice upto 2000Sqft. As the area gets bigger, there is not much increase in SalePrice.

The GarageCars capacity for cars has direct relationship with price. As the number of cars for garage increases upto 4, there is increase in price. As the number further increases upto 5, the SalePrice decreases.

4. EDA of categorical variables Categorical data

4a. Missing value treatment

```
categorical_train_df.isnull().sum(axis=0).sort_values(ascending=False)
```

	MSZoning	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType	HouseStyle	RoofStyle	RoofMatl	Exterior1st	Exterior2nd	ExterQual	ExterCond	Foundation	BsmtQual	BsmtCond	BsmtExposure	BsmtFinType1	BsmtFinType2	Heating	HeatingQC	CentralAir	Electrical	KitchenQual	Functional	FireplaceQu	GarageType	GarageFinish	GarageQual	GarageCond	PavedDrive	PoolQC	Fence	MiscFeature	SaleType	SaleCondition
0	RL	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	Norm	1Fam	2Story	Gable	CompShg	VinylSd	VinylSd	Gd	TA	PConc	Gd	TA	No	GLQ	Unf	GasA	Ex	Y	SBrkr	Gd	Typ	NaN	Attchd	RFn	TA	TA	Y	NaN	NaN	NaN	WD	Normal
1	RL	Pave	NaN	Reg	Lvl	AllPub	FR2	Gtl	Veenker	Feedr	Norm	1Fam	1Story	Gable	CompShg	MetalSd	MetalSd	TA	TA	CBlock	Gd	TA	Gd	ALQ	Rec	GasA	Ex	Y	SBrkr	TA	Typ	TA	Attchd	RFn	TA	TA	Y	NaN	NaN	NaN	WD	Abnorml
2	RL	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	CollgCr	Norm	Norm	1Fam	2Story	Gable	CompShg	VinylSd	VinylSd	Gd	TA	PConc	Gd	TA	Mn	GLQ	Unf	GasA	Ex	Y	SBrkr	Gd	Typ	TA	Attchd	RFn	TA	TA	Y	NaN	NaN	NaN	WD	Normal
3	RL	Pave	NaN	IR1	Lvl	AllPub	Corner	Gtl	Crawfor	Norm	Norm	1Fam	2Story	Gable	CompShg	Wd Sdng	Wd Shng	TA	TA	BrkTil	TA	Gd	No	ALQ	Unf	GasA	Gd	Y	SBrkr	Gd	Typ	Gd	Detchd	Unf	TA	TA	Y	NaN	NaN	NaN	WD	Abnorml
4	RL	Pave	NaN	IR1	Lvl	AllPub	FR2	Gtl	NoRidge	Norm	Norm	1Fam	2Story	Gable	CompShg	VinylSd	VinylSd	Gd	TA	PConc	Gd	TA	Av	GLQ	Unf	GasA	Ex	Y	SBrkr	Gd	Typ	TA	Attchd	RFn	TA	TA	Y	NaN	NaN	NaN	WD	Normal
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
1455	RL	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Gilbert	Norm	Norm	1Fam	2Story	Gable	CompShg	VinylSd	VinylSd	Gd	TA	PConc	Gd	TA	No	GLQ	Unf	GasA	Ex	Y	SBrkr	Gd	Typ	NaN	Attchd	RFn	TA	TA	Y	NaN	NaN	NaN	WD	Normal
1456	RL	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	NWAmes	Norm	Norm	1Fam	1Story	Gable	CompShg	Plywood	Plywood	TA	TA	CBlock	TA	TA	Mn	ALQ	Unf	GasA	TA	Y	SBrkr	TA	Min2	TA	Attchd	Unf	TA	TA	Y	NaN	NaN	NaN	WD	Abnorml
1457	RL	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Crawfor	Norm	Norm	1Fam	2Story	Gable	CompShg	CemntBd	CmentBd	Ex	TA	Stone	TA	TA	No	GLQ	Unf	GasA	Ex	Y	SBrkr	Gd	Typ	Gd	Attchd	RFn	TA	TA	Y	NaN	MnPrv	NaN	WD	Normal
1458	RL	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	NAmes	Norm	Norm	1Fam	1Story	Hip	CompShg	MetalSd	MetalSd	TA	TA	CBlock	TA	TA	No	BLQ	Unf	GasA	TA	Y	SBrkr	TA	Typ	TA	NaN	NaN	NaN	NaN	Y	NaN	NaN	NaN	WD	Normal
1459	RL	Pave	NaN	Reg	Lvl	AllPub	Inside	Gtl	Edwards	Norm	Norm	1Fam	1Story	Gable	CompShg	HdBoard	HdBoard	Gd	TA	PConc	Gd	TA	No	ALQ	Unf	GasA	TA	Y	SBrkr	TA	Typ	TA	Attchd	RFn	TA	TA	Y	NaN	NaN	NaN	WD	Normal

1460 rows × 37 columns

```
categorical_train_df[categorical_train_df.select_dtypes(exclude=['object'])].isnull().sum(axis=0)
```

```
Series([], dtype: int64)
```

```
categorical_train_df.describe(percentile=[.1,.2,.3,.4,.5,.6,.7,.8,.9,.99,.100])
```

	MSZoning	Street	Alley	LotShape	LandContour	Utilities	LotConfig	LandSlope	Neighborhood	Condition1	Condition2	BldgType	HouseStyle	RoofStyle	RoofMatl	Exterior1st	Exterior2nd	ExterQual	ExterCond	Foundation	BsmtQual	BsmtCond	BsmtExposure	BsmtFinType1	BsmtFinType2	Heating	HeatingQC	CentralAir	Electrical	KitchenQual	Functional	FireplaceQu	GarageType	GarageFinish	GarageQual	GarageCond	PavedDrive	PoolQC	Fence	MiscFeature	SaleType	SaleCondition
count	1460	1460	91	1460	1460	1460	1460	1460	1460	1460	1460	1460	1460	1460	1460	1460	1460	1460	1460	1460	1460	1460	1460	1460	1460	1460	1460	1460	1460	1460	1460	730	1460	1460	1460	1460	7	281	54	1460	1460
mean	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
std	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
min	NaN	NaN

```
Id             0
MSSubClass     0
MSZoning       0
LotFrontage    0
LotArea        0
Street         0
Alley          0
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
HeatingQC      0
CentralAir     0
Electrical     1
1stFlrSF       0
2ndFlrSF       0
LowQualFinSF   0
GrLivArea      0
BsmtFullBath   0
BsmtHalfBath   0
FullBath       0
HalfBath       0
BedroomAbvGr   0
KitchenAbvGr   0
KitchenQual    0
TotRmsAbvGrd   0
Functional     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
dtype: int64
```

```
[In 14]: pd.set_option('display.max_rows')
[In 15]: df[df.columns[df.isnull().sum()/df.shape[0]>.50]].isnull().sum()/df.shape[0]*100
```

```
Alley          93.767123
FireplaceQu    47.260274
PoolQC         99.520548
Fence          80.753425
MiscFeature    96.301370
dtype: float64
```

Alley , FireplaceQu, MiscFeature, Fence, PoolQC variables have > 50% null values.

```
[In 16]: df.drop(df[df.columns[df.isnull().sum()/df.shape[0]>.50]],axis=1,inplace=True)
```

```
[In 17]: df.shape
```

```
(1460, 75)
```

```
[In 18]: df.isnull().sum()
```

```
Id               0
MSSubClass       0
MSZoning         0
LotFrontage    259
LotArea          0
Street           0
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
HeatingQC        0
CentralAir       0
Electrical       1
1stFlrSF         0
2ndFlrSF         0
LowQualFinSF     0
GrLivArea        0
BsmtFullBath     0
BsmtHalfBath     0
FullBath         0
HalfBath         0
BedroomAbvGr     0
KitchenAbvGr     0
KitchenQual      0
TotRmsAbvGrd     0
Functional       0
Fireplaces       0
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
MiscVal          0
MoSold           0
YrSold           0
SaleType         0
SaleCondition    0
dtype: int64
```

```
[In 19]: df[df.columns[df.isnull().sum()/df.shape[0]>.0]].isnull().sum()/df.shape[0]*100
```

```
LotFrontage    17.739726
MasVnrType      0.547945
MasVnrArea      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
GarageYrBlt     5.547945
GarageFinish    5.547945
GarageQual      5.547945
GarageCond      5.547945
PoolQC         99.520548
Fence          80.753425
MiscFeature    96.301370
dtype: float64
```

Basement related variables are combined together in a new dataframe called basement_df. Then, using loc code look for rows where "TotalBsmtSF" is zero.

The basement related variables are filled with NA wherever "TotalBsmtSF" is zero.

```
[In 20]: basement_df=df[['Id','BsmtQual','BsmtCond','BsmtExposure','BsmtFinType1','BsmtFinSF1','BsmtFinType2','BsmtFinSF2','BsmtUnfSF','TotalBsmtSF']]
```

```
[In 21]: basement_df.shape
```

```
(1460, 10)
```

```
[In 22]: basement_df.isnull().sum()/basement_df.shape[0]*100
```

```
Id               0.000000
BsmtQual         2.534247
BsmtCond         2.534247
BsmtExposure     2.602740
BsmtFinType1     2.534247
BsmtFinSF1       0.000000
BsmtFinType2     2.602740
BsmtFinSF2       0.000000
BsmtUnfSF        0.000000
TotalBsmtSF      0.000000
dtype: float64
```

```
[In 23]: basement_df[basement_df.TotalBsmtSF==0]
```

	Id	BsmtQual	BsmtCond	BsmtExposure	BsmtFinType1	BsmtFinSF1	BsmtFinType2	BsmtFinSF2	BsmtUnfSF	TotalBsmtSF
39	40	NaN	NaN	NaN	NaN	0	NaN	0	0	0
48	49	Ex	Po	No	GLQ	455	Unf	0	290	745
90	91	NaN	NaN	NaN	NaN	0	NaN	0	0	0
102	103	NaN	NaN	NaN	NaN	0	NaN	0	0	0
227	228	NaN	NaN	NaN	NaN	0	NaN	0	0	0
232	233	NaN	NaN	NaN	NaN	0	NaN	0	0	0
239	240	NaN	NaN	NaN	NaN	0	NaN	0	0	0
279	280	NaN	NaN	NaN	NaN	0	NaN	0	0	0
332	333	NaN	NaN	NaN	NaN	0	NaN	0	0	0
343	344	NaN	NaN	NaN	NaN	0	NaN	0	0	0
363	364	NaN	NaN	NaN	NaN	0	NaN	0	0	0
371	372	NaN	NaN	NaN	NaN	0	NaN	0	0	0
390	391	NaN	NaN	NaN	NaN	0	NaN	0	0	0
421	422	NaN	NaN	NaN	NaN	0	NaN	0	0	0
444	445	NaN	NaN	NaN	NaN	0	NaN	0	0	0
485	486	NaN	NaN	NaN	NaN	0	NaN	0	0	0
490	491	NaN	NaN	NaN	NaN	0	NaN	0	0	0
494	495	NaN	NaN	NaN	NaN	0	NaN	0	0	0
517	518	NaN	NaN	NaN	NaN	0	NaN	0	0	0
533	534	NaN	NaN	NaN	NaN	0	NaN	0	0	0
548	549	NaN	NaN	NaN	NaN	0	NaN	0	0	0
580	581	NaN	NaN	NaN	NaN	0	NaN	0	0	0
725	726	NaN	NaN	NaN	NaN	0	NaN	0	0	0
757	758	NaN	NaN	NaN	NaN	0	NaN	0	0	0
758	759	NaN	NaN	NaN	NaN	0	NaN	0	0	0
773	774	NaN	NaN	NaN	NaN	0	NaN	0	0	0
874	875	NaN	NaN	NaN	NaN	0	NaN	0	0	0
888	889	NaN	NaN	NaN	NaN	0	NaN	0	0	0
1011	1012	NaN	NaN	NaN	NaN	0	NaN	0	0	0
1035	1036	NaN	NaN	NaN	NaN	0	NaN	0	0	0
1045	1046	NaN	NaN	NaN	NaN	0	NaN	0	0	0
1048	1049	NaN	NaN	NaN	NaN	0	NaN	0	0	0
1049	1050	Ex	Po	No	GLQ	1124	Unf	0	479	1603
1090	1091	NaN	NaN	NaN	NaN	0	NaN	0	0	0
1092	1093	NaN	NaN	NaN	NaN	0	NaN	0	0	0
1107	1108	NaN	NaN	NaN	NaN	0	NaN	0	0	0
1123	1124	NaN	NaN	NaN	NaN	0	NaN	0	0	0
1132	1133	NaN	NaN	NaN	NaN	0	NaN	0	0	0
1137	1138	NaN	NaN	NaN	NaN	0	NaN	0	0	0
1216	1217	NaN	NaN	NaN	NaN	0	NaN	0	0	0
1232	1233	NaN	NaN	NaN	NaN	0	NaN	0	0	0
1321	1322	NaN	NaN	NaN	NaN	0	NaN	0	0	0
1335	1336	NaN	NaN	NaN	NaN	0	NaN	0	0	0
1350	1351	NaN	NaN	NaN	NaN	0	NaN	0	0	0

```
[In 24]: basement_df.loc[basement_df.TotalBsmtSF==0,['BsmtQual','BsmtCond','BsmtExposure','BsmtFinType1','BsmtFinSF1','BsmtFinType2','BsmtFinSF2','BsmtUnfSF','TotalBsmtSF']]=np.NaN
```

Garage related variables are combined together in a new dataframe called garage_df. Then, using loc code look for rows where "GarageCars" is zero.

The garage related variables are filled with NA.

```
[In 25]: garage_df=df[['Id','GarageType','GarageYrBlt','GarageFinish','GarageCars','GarageArea','GarageQual','GarageCond']]
```

```
[In 26]: garage_df.shape
```

```
(1460, 8)
```

```
[In 27]: garage_df.isnull().sum()/garage_df.shape[0]*100
```

```
Id             0.000000
GarageType     5.547945
GarageYrBlt    5.547945
GarageFinish   5.547945
GarageCars     0.000000
GarageArea     0.000000
GarageQual     5.547945
GarageCond     5.547945
dtype: float64
```

```
[In 28]: garage_df[garage_df.GarageCars==0]
```

	Id	GarageType	GarageYrBlt	GarageFinish	GarageCars	GarageArea	GarageQual	GarageCond
39	40	Detchd	NaN	Unf	0	0	Po	Po
48	49	NaN	NaN	NaN	0	0	NaN	NaN
78	79	NaN	NaN	NaN	0	0	NaN	NaN
88	89	Detchd	NaN	Unf	0	0	Po	Po
90	91	NaN	NaN	NaN	0	0	NaN	NaN
99	100	Detchd	NaN	Unf	0	0	Po	Po
102	103	NaN	NaN	NaN	0	0	NaN	NaN
203	204	NaN	NaN	NaN	0	0	NaN	NaN
227	228	NaN	NaN	NaN	0	0	NaN	NaN
232	233	NaN	NaN	NaN	0	0	NaN	NaN
239	240	NaN	NaN	NaN	0	0	NaN	NaN
279	280	NaN	NaN	NaN	0	0	NaN	NaN
332	333	NaN	NaN	NaN	0	0	NaN	NaN
343	344	NaN	NaN	NaN	0	0	NaN	NaN
363	364	NaN	NaN	NaN	0	0	NaN	NaN
371	372	NaN	NaN	NaN	0	0	NaN	NaN
390	391	NaN	NaN	NaN	0	0	NaN	NaN
421	422	NaN	NaN	NaN	0	0	NaN	NaN
444	445	NaN	NaN	NaN	0	0	NaN	NaN
485	486	NaN	NaN	NaN	0	0	NaN	NaN
490	491	NaN	NaN	NaN	0	0	NaN	NaN
494	495	NaN	NaN	NaN	0	0	NaN	NaN
517	518	NaN	NaN	NaN	0	0	NaN	NaN
533	534	NaN	NaN	NaN	0	0	NaN	NaN
548	549	NaN	NaN	NaN	0	0	NaN	NaN
580	581	NaN	NaN	NaN	0	0	NaN	NaN
725	726	NaN	NaN	NaN	0	0	NaN	NaN
757	758	NaN	NaN	NaN	0	0	NaN	NaN
758	759	NaN	NaN	NaN	0	0	NaN	NaN
773	774	NaN	NaN	NaN	0	0	NaN	NaN
874	875	NaN	NaN	NaN	0	0	NaN	NaN
888	889	NaN	NaN	NaN	0	0	NaN	NaN
1011	1012	NaN	NaN	NaN	0	0	NaN	NaN
1035	1036	NaN	NaN	NaN	0	0	NaN	NaN
1045	1046	NaN	NaN	NaN	0	0	NaN	NaN
1048	1049	NaN	NaN	NaN	0	0	NaN	NaN
1049	1050	Detchd	1923	Unf	0	864	TA	TA
1090	1091	NaN	NaN	NaN	0	0	NaN	NaN
1092	1093	NaN	NaN	NaN	0	0	NaN	NaN
1107	1108	NaN	NaN	NaN	0	0	NaN	NaN
1123	1124	NaN	NaN	NaN	0	0	NaN	NaN
1132	1133	NaN	NaN	NaN	0	0	NaN	NaN
1137	1138	NaN	NaN	NaN	0	0	NaN	NaN
1216	1217	NaN	NaN	NaN	0	0	NaN	NaN
1232	1233	NaN	NaN	NaN	0	0	NaN	NaN
1321	1322	NaN	NaN	NaN	0	0	NaN	NaN
1335	1336	NaN	NaN	NaN	0	0	NaN	NaN
1350	1351	NaN	NaN	NaN	0	0	NaN	NaN

```
[In 29]: garage_df.loc[garage_df.GarageCars==0,['GarageType','GarageYrBlt','GarageFinish','GarageCars','GarageArea','GarageQual','GarageCond']]=np.NaN
```

```
[In 30]: df[df.columns[df.isnull().sum()/df.shape[0]>.0]].isnull().sum()/df.shape[0]*100
```

```
LotFrontage    17.739726
MasVnrType      0.547945
MasVnrArea      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
GarageYrBlt     5.547945
GarageFinish    5.547945
GarageQual      5.547945
GarageCond      5.547945
PoolQC         99.520548
Fence          80.753425
MiscFeature    96.301370
dtype: float64
```

The categorical data does not show any null values after missing value treatment

do Count plot and box plot for bivariate analysis

```
[In 31]: df[df.columns[df.isnull().sum()/df.shape[0]==0]].shape
```

```
(1460, 62)
```

```
[In 32]: df[df.columns[df.isnull().sum()/df.shape[0]==0]].columns
```

```
Index(['Id', 'MSSubClass', 'MSZoning', 'LotArea', 'Street', 'LotShape',
       'LandContour', 'Utilities', 'LotConfig', 'LandSlope', 'Neighborhood',
       'Condition1', 'Condition2', 'BldgType', 'HouseStyle', 'OverallQual',
       'OverallCond', 'YearBuilt', 'YearRemodAdd', 'RoofStyle', 'RoofMatl',
       'Exterior1st', 'Exterior2nd', 'ExterQual', 'ExterCond', 'Foundation',
       'Heating', 'HeatingQC', 'CentralAir', '1stFlrSF', '2ndFlrSF',
       'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath',
       'FullBath', 'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual',
       'TotRmsAbvGrd', 'Functional', 'Fireplaces', 'PavedDrive',
       'WoodDeckSF', 'OpenPorchSF', 'EnclosedPorch', '3SsnPorch',
       'ScreenPorch', 'PoolArea', 'MiscVal', 'MoSold', 'YrSold',
       'SaleType', 'SaleCondition', 'SalePrice'],
      dtype='object')
```

```
[In 33]: df[df.columns[df.isnull().sum()/df.shape[0]==0]].dtypes
```

```
Id               int64
MSSubClass       int64
MSZoning        object
LotArea          int64
Street          object
LotShape        object
LandContour     object
Utilities       object
LotConfig       object
LandSlope       object
Neighborhood    object
Condition1      object
Condition2      object
BldgType        object
HouseStyle      object
OverallQual      int64
OverallCond      int64
YearBuilt        int64
YearRemodAdd     int64
RoofStyle       object
RoofMatl        object
Exterior1st     object
Exterior2nd     object
ExterQual       object
ExterCond       object
Foundation      object
Heating         object
HeatingQC       object
CentralAir      object
1stFlrSF         int64
2ndFlrSF         int64
LowQualFinSF     int64
GrLivArea        int64
BsmtFullBath     int64
BsmtHalfBath     int64
FullBath         int64
HalfBath         int64
BedroomAbvGr     int64
KitchenAbvGr     int64
KitchenQual     object
TotRmsAbvGrd     int64
Functional      object
Fireplaces       int64
PavedDrive      object
WoodDeckSF       int64
OpenPorchSF      int64
EnclosedPorch    int64
3SsnPorch        int64
ScreenPorch      int64
PoolArea         int64
MiscVal          int64
MoSold           int64
YrSold           int64
SaleType        object
SaleCondition   object
SalePrice        int64
dtype: int64
```

```
[In 34]: df[df.columns[df.isnull().sum()/df.shape[0]==0]].shape[1]-1
```

```
61
```

```
[In 35]: for i in df[df.columns[df.isnull().sum()/df.shape[0]==0]].columns:
         ax=plt.figure(figsize=(8,4))
         ax=sns.countplot(x=i,data=df)
         ax=plt.xticks(rotation=90)
         ax=plt.title('Count Plot for '+i+' Variable')
         plt.show()
         ax=sns.boxplot(x=df.SalePrice,y=df[i],data=df)
         ax=plt.title('Box Plot for '+i+' Variable')
         plt.show
```

MSZoning: The rate price is higher for houses built in Residential Low Density (RL), Floating Village Residential (FV) zones compare to other zoning areas.

MSZoning: Although RL has many outliers ,The houses built in commercial are have lowest median SalePrice of that house by 100000 compare to other zoning areas.

Street: The SalePrice is higher in case Paw type street compare to Grvl. Therefore SalePrice is dependent on street type.

BldgType: The single family house sells higher than any other category. However, the SalePrice is dependent on the building type.

KitchenQual: The graph shows that Kitchen with excellent quality increases median SalePrice of that house by 100000 compare to kitchen with good quality

GarageFinish: The houses that finished garage sell higher than other categories. The SalePrice of the houses with unfinished and no garage is significantly lower than finished garage.

SaleType: The SalePrice is clearly higher for newly built home compare to other categories. SalePrice is dependent on SaleType.

```
[In 36]: df[df.columns[df.isnull().sum()/df.shape[0]==0]].shape[1]-1
```

```
61
```

```
[In 37]: for i in df[df.columns[df.isnull().sum()/df.shape[0]==0]].columns:
         ax=plt.figure(figsize=(8,4))
         ax=sns.scatplot(x=i,y=df.SalePrice,data=df)
         ax=plt.title('Scatplot for '+i+' Variable')
         plt.show
```

```
[In 38]: df[df.columns[df.isnull().sum()/df.shape[0]==0]].shape[1]-1
```

```
61
```

```
[In 39]: df.to_csv('Cleaned_Data.csv')
```

Street: The graph shows that most house are built with three street type with more than 1400 records.

LotShape: The graph shows that mostly house lot is regular shaped (1,000) or slightly irregular (450).

BldgType : The graph shows that most houses in the market are single family houses (nearly 1500).

KitchenQual: The graph shows that kitchen quality is mostly typical or good.

BsmtQual : The graph shows that mostly houses are built with typical and good condition basement.

ie. Identify significant variable using p-value and Chi Square values.

```
import scipy.stats as st
for i in cat_cols:
    print('Variable: {}'.format(i))
    print('Chi Square Test of Independence')
    print('Null Hypothesis: Variable and SalePrice are independent')
    print('Alternate Hypothesis: Variable and SalePrice are not independent')
    print('Level of Significance: 5%')
    print('Degree of Freedom: 1')
    print('Critical Value: 3.841')
    print('Calculated Value: {}'.format(round(st.chi2_contingency(
        pd.crosstab(df[i],df['SalePrice'])
        )[0],2)))
    print('Calculated Value: {}'.format(round(st.chi2_contingency(
        pd.crosstab(df[i],df['SalePrice'])
        )[1],2)))
    print('Decision: REJECTED')
    print('Conclusion: {} is an important predictor, Discard it from model'.format(i))
    print('\n')
```