In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:
```
df=pd.read_csv("F:\DATA SCIENCE PROGRAM\Datascience with Python\Datasets\OSL Datase
```

In [4]:
```
df
```

Out[4]:

	Unnamed: 0	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContou
0	0	SC60	RL	65	8450	Pave	None	Reg	Lv
1	1	SC20	RL	80	9600	Pave	None	Reg	Lv
2	2	SC60	RL	68	11250	Pave	None	IR1	Lv
3	3	SC70	RL	60	9550	Pave	None	IR1	Lv
4	4	SC60	RL	84	14260	Pave	None	IR1	Lv
...	...	...	...	...	...	...	...	...	.
1455	1455	SC60	RL	62	7917	Pave	None	Reg	Lv
1456	1456	SC20	RL	85	13175	Pave	None	Reg	Lv
1457	1457	SC70	RL	66	9042	Pave	None	Reg	Lv
1458	1458	SC20	RL	68	9717	Pave	None	Reg	Lv
1459	1459	SC20	RL	75	9937	Pave	None	Reg	Lv

1460 rows × 81 columns

In [5]:
```
#identify the shape

df.shape
```

Out[5]:
```
(1460, 81)
```

In [6]:
```
#identify the null values
df.isna().sum()
```

Out[6]:
```
Unnamed: 0       0
MSSubClass       0
MSZoning         0
LotFrontage      0
LotArea          0
                ..
MoSold           0
YrSold           0
SaleType         0
SaleCondition    0
SalePrice        0
Length: 81, dtype: int64
```

In [7]:
```
#identify the variables with unique values
df.columns
```

Out[7]:
```
Index(['Unnamed: 0', 'MSSubClass', 'MSZoning', 'LotFrontage', 'LotArea',
       'Street', 'Alley', 'LotShape', 'LandContour', 'Utilities', 'LotConfig',
       'LandSlope', 'Neighborhood', 'Condition1', 'Condition2', 'BldgType',
       'HouseStyle', 'OverallQual', 'OverallCond', 'YearBuilt', 'YearRemodAdd',
       'RoofStyle', 'RoofMatl', 'Exterior1st', 'Exterior2nd', 'MasVnrType',
       'MasVnrArea', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual',
       'BsmtCond', 'BsmtExposure', 'BsmtFinType1', 'BsmtFinSF1',
       'BsmtFinType2', 'BsmtFinSF2', 'BsmtUnfSF', 'TotalBsmtSF', 'Heating',
       'HeatingQC', 'CentralAir', 'Electrical', '1stFlrSF', '2ndFlrSF',
       'LowQualFinSF', 'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath',
       'HalfBath', 'BedroomAbvGr', 'KitchenAbvGr', 'KitchenQual',
       'TotRmsAbvGrd', 'Functional', 'Fireplaces', 'FireplaceQu', 'GarageType',
       'GarageYrBlt', 'GarageFinish', 'GarageCars', 'GarageArea', 'GarageQual',
       'GarageCond', 'PavedDrive', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'PoolQC',
       'Fence', 'MiscFeature', 'MiscVal', 'MoSold', 'YrSold', 'SaleType',
       'SaleCondition', 'SalePrice'],
      dtype='object')
```

In [8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Unnamed: 0     1460 non-null   int64
 1   MSSubClass     1460 non-null   object
 2   MSZoning       1460 non-null   object
 3   LotFrontage    1460 non-null   int64
 4   LotArea        1460 non-null   int64
 5   Street         1460 non-null   object
 6   Alley          1460 non-null   object
 7   LotShape       1460 non-null   object
 8   LandContour    1460 non-null   object
 9   Utilities      1460 non-null   object
 10  LotConfig      1460 non-null   object
 11  LandSlope      1460 non-null   object
 12  Neighborhood   1460 non-null   object
 13  Condition1     1460 non-null   object
 14  Condition2     1460 non-null   object
 15  BldgType       1460 non-null   object
 16  HouseStyle     1460 non-null   object
 17  OverallQual    1460 non-null   int64
 18  OverallCond    1460 non-null   int64
 19  YearBuilt      1460 non-null   int64
 20  YearRemodAdd   1460 non-null   int64
 21  RoofStyle      1460 non-null   object
 22  RoofMatl       1460 non-null   object
 23  Exterior1st    1460 non-null   object
 24  Exterior2nd    1460 non-null   object
 25  MasVnrType     1460 non-null   object
 26  MasVnrArea     1460 non-null   int64
 27  ExterQual      1460 non-null   object
 28  ExterCond      1460 non-null   object
 29  Foundation     1460 non-null   object
 30  BsmtQual       1460 non-null   object
 31  BsmtCond       1460 non-null   object
 32  BsmtExposure   1460 non-null   object
 33  BsmtFinType1   1460 non-null   object
 34  BsmtFinSF1     1460 non-null   int64
 35  BsmtFinType2   1460 non-null   object
 36  BsmtFinSF2     1460 non-null   int64
 37  BsmtUnfSF      1460 non-null   int64
 38  TotalBsmtSF    1460 non-null   int64
 39  Heating        1460 non-null   object
 40  HeatingQC      1460 non-null   object
 41  CentralAir     1460 non-null   object
 42  Electrical     1459 non-null   object
 43  1stFlrSF       1460 non-null   int64
 44  2ndFlrSF       1460 non-null   int64
 45  LowQualFinSF   1460 non-null   int64
 46  GrLivArea      1460 non-null   int64
 47  BsmtFullBath   1460 non-null   int64
 48  BsmtHalfBath   1460 non-null   int64
 49  FullBath       1460 non-null   int64
 50  HalfBath       1460 non-null   int64
 51  BedroomAbvGr   1460 non-null   int64
```

```
 52  KitchenAbvGr   1460 non-null   int64
 53  KitchenQual    1460 non-null   object
 54  TotRmsAbvGrd   1460 non-null   int64
 55  Functional     1460 non-null   object
 56  Fireplaces     1460 non-null   int64
 57  FireplaceQu    1460 non-null   object
 58  GarageType     1460 non-null   object
 59  GarageYrBlt    1379 non-null   float64
 60  GarageFinish   1460 non-null   object
 61  GarageCars     1460 non-null   int64
 62  GarageArea     1460 non-null   int64
 63  GarageQual     1460 non-null   object
 64  GarageCond     1460 non-null   object
 65  PavedDrive     1460 non-null   object
 66  WoodDeckSF     1460 non-null   int64
 67  OpenPorchSF    1460 non-null   int64
 68  EnclosedPorch  1460 non-null   int64
 69  3SsnPorch      1460 non-null   int64
 70  ScreenPorch    1460 non-null   int64
 71  PoolArea       1460 non-null   int64
 72  PoolQC         1460 non-null   object
 73  Fence          1460 non-null   object
 74  MiscFeature    1460 non-null   object
 75  MiscVal        1460 non-null   int64
 76  MoSold         1460 non-null   object
 77  YrSold         1460 non-null   int64
 78  SaleType       1460 non-null   object
 79  SaleCondition  1460 non-null   object
 80  SalePrice      1460 non-null   int64
dtypes: float64(1), int64(35), object(45)
memory usage: 924.0+ KB
```

Generate a separate dataset for numerical and categorical variables

In [9]:
```
df_numerical=(df._get_numeric_data())
df_numerical.columns
```

Out[9]:
```
Index(['Unnamed: 0', 'LotFrontage', 'LotArea', 'OverallQual', 'OverallCond',
       'YearBuilt', 'YearRemodAdd', 'MasVnrArea', 'BsmtFinSF1', 'BsmtFinSF2',
       'BsmtUnfSF', 'TotalBsmtSF', '1stFlrSF', '2ndFlrSF', 'LowQualFinSF',
       'GrLivArea', 'BsmtFullBath', 'BsmtHalfBath', 'FullBath', 'HalfBath',
       'BedroomAbvGr', 'KitchenAbvGr', 'TotRmsAbvGrd', 'Fireplaces',
       'GarageYrBlt', 'GarageCars', 'GarageArea', 'WoodDeckSF', 'OpenPorchSF',
       'EnclosedPorch', '3SsnPorch', 'ScreenPorch', 'PoolArea', 'MiscVal',
       'YrSold', 'SalePrice'],
      dtype='object')
```

In [10]:
```
df_categorical=(set(df)-set(df._get_numeric_data()))
df_categorical
```

Out[10]:
```
{'Alley',
 'BldgType',
 'BsmtCond',
 'BsmtExposure',
 'BsmtFinType1',
 'BsmtFinType2',
 'BsmtQual',
 'CentralAir',
 'Condition1',
 'Condition2',
 'Electrical',
 'ExterCond',
 'ExterQual',
 'Exterior1st',
 'Exterior2nd',
 'Fence',
 'FireplaceQu',
 'Foundation',
 'Functional',
 'GarageCond',
 'GarageFinish',
 'GarageQual',
 'GarageType',
 'Heating',
 'HeatingQC',
 'HouseStyle',
 'KitchenQual',
 'LandContour',
 'LandSlope',
 'LotConfig',
 'LotShape',
 'MSSubClass',
 'MSZoning',
 'MasVnrType',
 'MiscFeature',
 'MoSold',
 'Neighborhood',
 'PavedDrive',
 'PoolQC',
 'RoofMatl',
 'RoofStyle',
 'SaleCondition',
 'SaleType',
 'Street',
 'Utilities'}
```

Exploratory data analysis on numerical data

Missing value treatment

In [11]:
```
df_numerical.isna().sum()
```

Out[11]:
```
Unnamed: 0        0
LotFrontage       0
LotArea           0
OverallQual       0
OverallCond       0
YearBuilt         0
YearRemodAdd      0
MasVnrArea        0
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
YrSold            0
SalePrice         0
dtype: int64
```

In [12]:
```
df_numerical['GarageYrBlt'].describe()
```

Out[12]:
```
count    1379.000000
mean     1978.506164
std        24.689725
min      1900.000000
25%      1961.000000
50%      1980.000000
75%      2002.000000
max      2010.000000
Name: GarageYrBlt, dtype: float64
```

In [13]:
```
df_numerical['GarageYrBlt'].fillna(1978.5061,inplace=True)
```

In [14]:
```
df_numerical.isna().sum()
```

Out[14]:
```
Unnamed: 0       0
LotFrontage      0
LotArea          0
OverallQual      0
OverallCond      0
YearBuilt        0
YearRemodAdd     0
MasVnrArea       0
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      0
GarageCars       0
GarageArea       0
WoodDeckSF       0
OpenPorchSF      0
EnclosedPorch    0
3SsnPorch        0
ScreenPorch      0
PoolArea         0
MiscVal          0
YrSold           0
SalePrice        0
dtype: int64
```

In [15]:
```
#Identifying the skewness and distribution
df_numerical.describe()
```

Out[15]:

SF	EnclosedPorch	3SsnPorch	ScreenPorch	PoolArea	MiscVal	YrSold	SalePrice
00	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000	1460.000000
74	21.954110	3.409589	15.060959	2.758904	43.489041	2007.815753	180921.195890
28	61.119149	29.317331	55.757415	40.177307	496.123024	1.328095	79442.502883
00	0.000000	0.000000	0.000000	0.000000	0.000000	2006.000000	34900.000000
00	0.000000	0.000000	0.000000	0.000000	0.000000	2007.000000	129975.000000
00	0.000000	0.000000	0.000000	0.000000	0.000000	2008.000000	163000.000000
00	0.000000	0.000000	0.000000	0.000000	0.000000	2009.000000	214000.000000
00	552.000000	508.000000	480.000000	738.000000	15500.000000	2010.000000	755000.000000

In [16]:
```
import seaborn as sns
%matplotlib.inline
```

```
UsageError: Line magic function `%matplotlib.inline` not found.
```

b. Identify the skewness and distribution

In [17]:
```
df.skew(axis=0,skipna=True)
```

```
C:\Users\Dell\AppData\Local\Temp\ipykernel_7280\4266299306.py:1: FutureWarning: D
ropping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is
deprecated; in a future version this will raise TypeError.  Select only valid col
umns before calling the reduction.
  df.skew(axis=0,skipna=True)
```

Out[17]:
```
Unnamed: 0        0.000000
LotFrontage       0.267822
LotArea          12.207688
OverallQual       0.216944
OverallCond       0.693067
YearBuilt        -0.613461
YearRemodAdd     -0.503562
MasVnrArea        2.677616
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      -0.668174
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
YrSold            0.096269
SalePrice         1.882876
dtype: float64
```

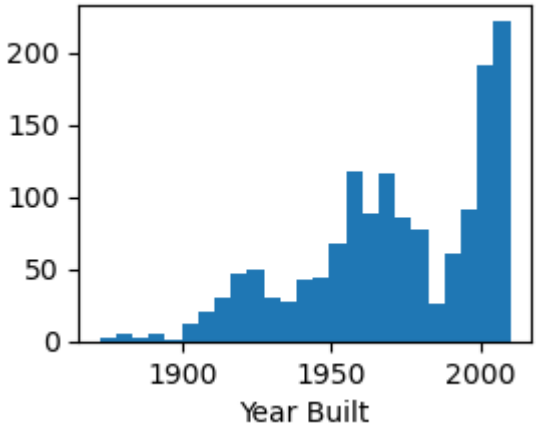
In [18]:
```
print("Skewness: %f" % df['SalePrice'].skew())
```

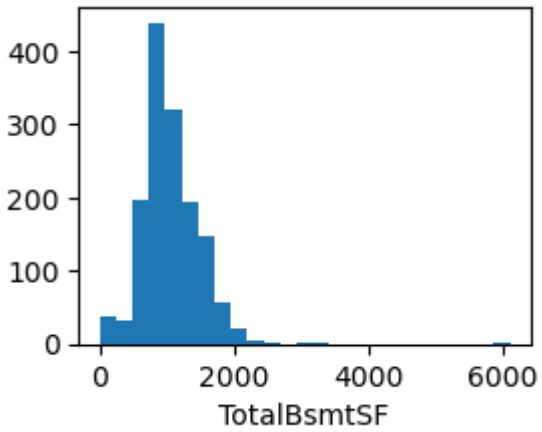
```
Skewness: 1.882876
```

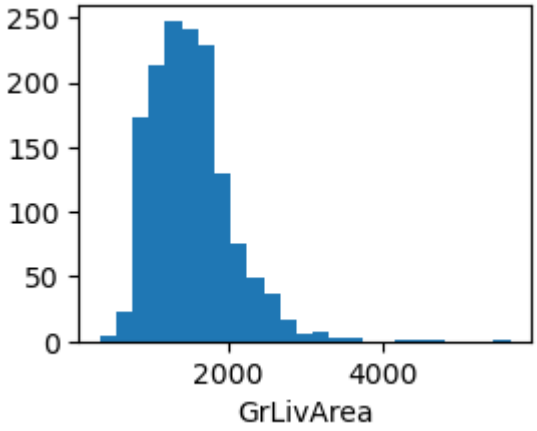
In [19]:
```
num_col=['YearBuilt', 'TotalBsmtSF', 'GrLivArea', 'SalePrice']
plt.figure(figsize=(10,10))
plt.subplots_adjust(hspace=0.9,wspace=0.5)
```

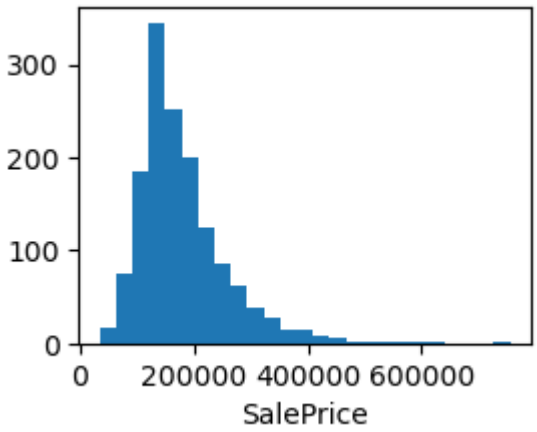
```
<Figure size 1000x1000 with 0 Axes>
```

In [20]:
```
plt.subplot(2,2,1)
plt.hist(df_numerical['YearBuilt'],bins=25)
plt.xlabel('Year Built')
plt.show()

plt.subplot(2,2,2)
plt.hist(df_numerical['TotalBsmtSF'],bins=25)
plt.xlabel('TotalBsmtSF')
plt.show()

plt.subplot(2,2,3)
plt.hist(df_numerical['GrLivArea'],bins=25)
plt.xlabel('GrLivArea')
plt.show()

plt.subplot(2,2,4)
plt.hist(df_numerical['SalePrice'],bins=25)
plt.xlabel('SalePrice')
plt.show()
```









In [ ]:
```
plt.figure(figsize=(10,10))
plt.subplots_adjust(hspace=0.9,wspace=0.5)
facet= None

plt.subplot(2,2,1)
sns.boxplot(facet,df_numerical['YearBuilt'],data=df)
plt.show()

plt.subplot(2,2,2)
sns.boxplot(facet,df_numerical['TotalBsmtSF'],data=df)
plt.show()

plt.subplot(2,2,3)
sns.boxplot(facet,df_numerical['GrLivArea'],data=df)
plt.show()

plt.subplot(2,2,4)
sns.boxplot(facet,df_numerical['SalePrice'],data=df)
plt.show()
```

Comment- As we can see, they are many outliers in the columns with numerical data but the outliers are just skewed data and not really misinterprited data.

c. Identify significant variables using a correlation matrix

In [ ]:
```
#Lets plot the graph to check the correlation bet 'YearBuilt', 'OverallCond', 'GrLi
plt.figure(figsize=(5,5))
plt.subplots_adjust(hspace=0.9,wspace=0.5)



sns.scatterplot('YearBuilt','SalePrice',data=df)
plt.show()
```

In [ ]:
```
plt.figure(figsize=(5,5))
sns.scatterplot('OverallCond','SalePrice',data=df)
plt.show()
```

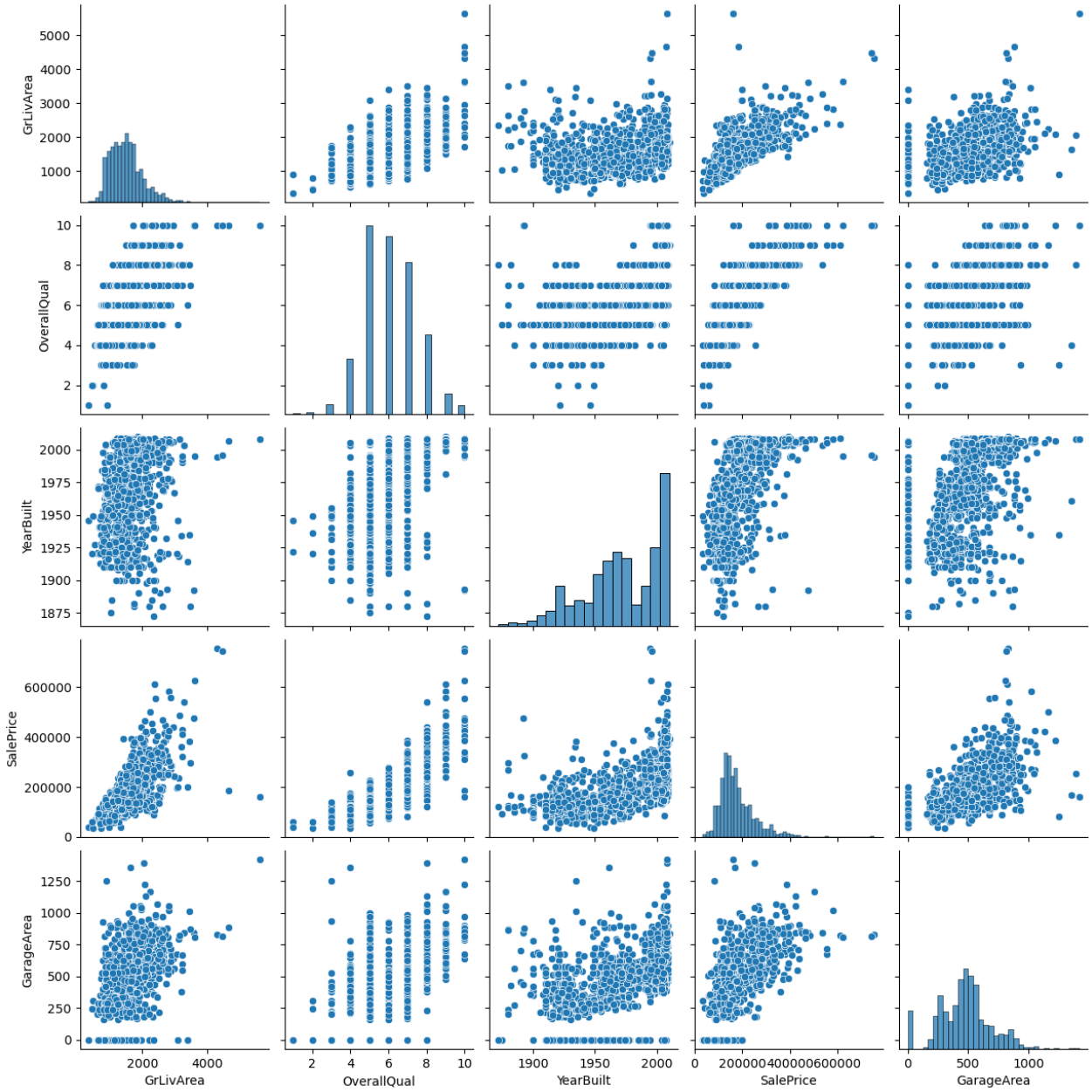
Comment - The salesprice is not directly proportional to the condition of the property because, the properties with average condition rating i.e. 5 have marked most sales and highest saleprice than the properties with excellent ratings.

In [ ]:
```
plt.figure(figsize=(5,5))
sns.scatterplot('GrLivArea','SalePrice',data=df)
plt.show()
```

d. Pair plot for distribution and density

In [29]:
```
columns=['GrLivArea','OverallQual','YearBuilt','SalePrice','GarageArea']
sns.pairplot(df[columns])
```

Out[29]: <seaborn.axisgrid.PairGrid at 0x270ea45c5b0>



Comment -

1. Age of the building is proportional to its selling prices.
2. Quality of the building has added to their selling prices.

4. EDA of categorical variables

a. Missing value treatment

In [22]:
```
df.isnull().sum()[df.isnull().sum()>0]
```

Out[22]:
```
Electrical    1
dtype: int64
```

In [23]:
```
x='Sbrkr'
df['Electrical'].fillna(x,inplace=True)
```

In [24]:
```
df['Electrical'].isna().any()
```

Out[24]:
```
False
```

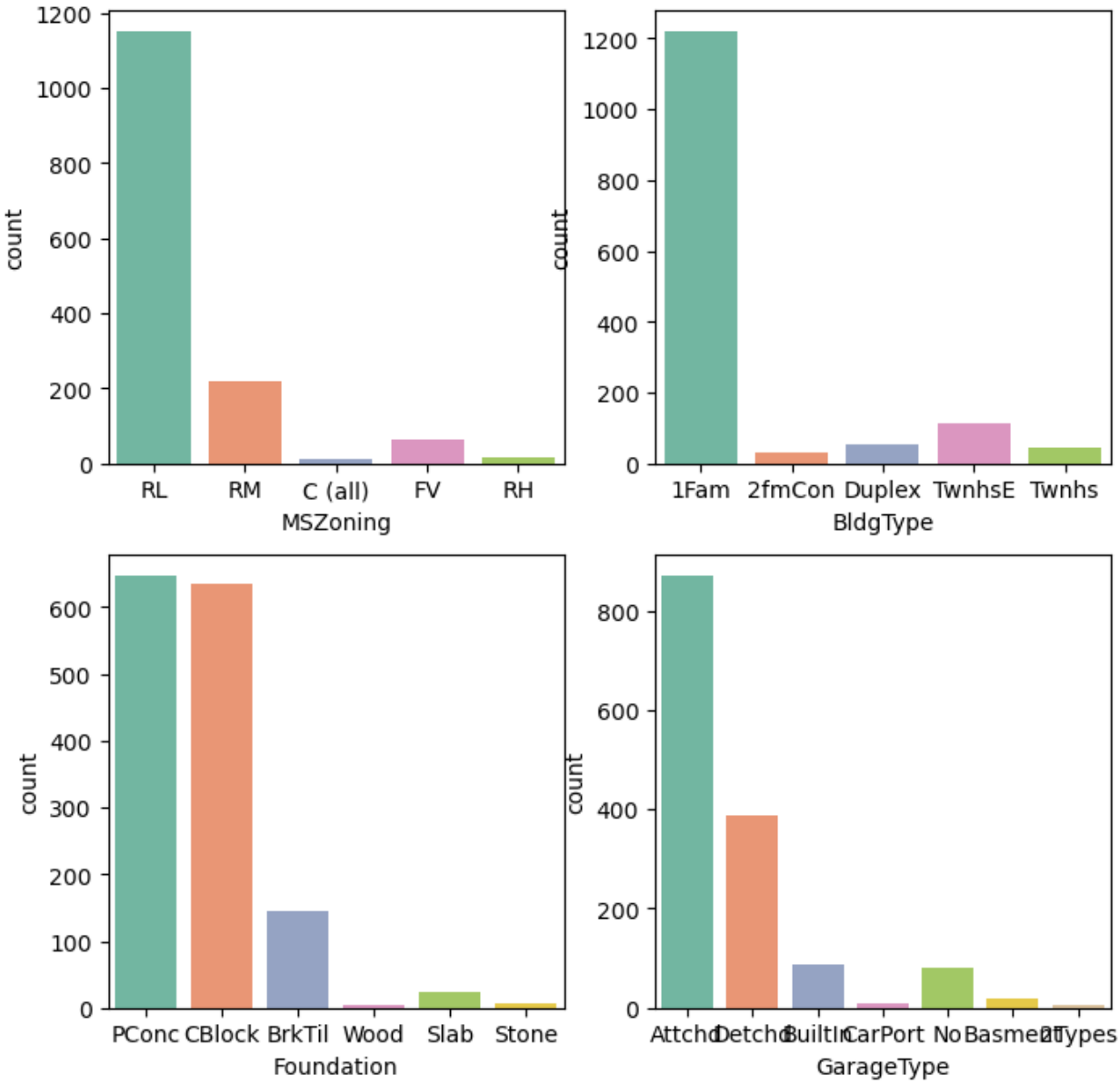
b. Count plot and box plot for bivariate analysis

In [25]:
```
df_categorical
```

Out[25]:
```
{'Alley',
 'BldgType',
 'BsmtCond',
 'BsmtExposure',
 'BsmtFinType1',
 'BsmtFinType2',
 'BsmtQual',
 'CentralAir',
 'Condition1',
 'Condition2',
 'Electrical',
 'ExterCond',
 'ExterQual',
 'Exterior1st',
 'Exterior2nd',
 'Fence',
 'FireplaceQu',
 'Foundation',
 'Functional',
 'GarageCond',
 'GarageFinish',
 'GarageQual',
 'GarageType',
 'Heating',
 'HeatingQC',
 'HouseStyle',
 'KitchenQual',
 'LandContour',
 'LandSlope',
 'LotConfig',
 'LotShape',
 'MSSubClass',
 'MSZoning',
 'MasVnrType',
 'MiscFeature',
 'MoSold',
 'Neighborhood',
 'PavedDrive',
 'PoolQC',
 'RoofMatl',
 'RoofStyle',
 'SaleCondition',
 'SaleType',
 'Street',
 'Utilities'}
```

In [26]:
```
plt.figure(figsize=(8,8))

plt.subplot(2,2,1)
sns.countplot(x=df['MSZoning'],palette='Set2')


plt.subplot(2,2,2)
sns.countplot(x=df['BldgType'],palette='Set2')


plt.subplot(2,2,3)
sns.countplot(x=df['Foundation'],palette='Set2')

plt.subplot(2,2,4)
sns.countplot(x=df['GarageType'],palette='Set2')
```

Out[26]: <AxesSubplot:xlabel='GarageType', ylabel='count'>

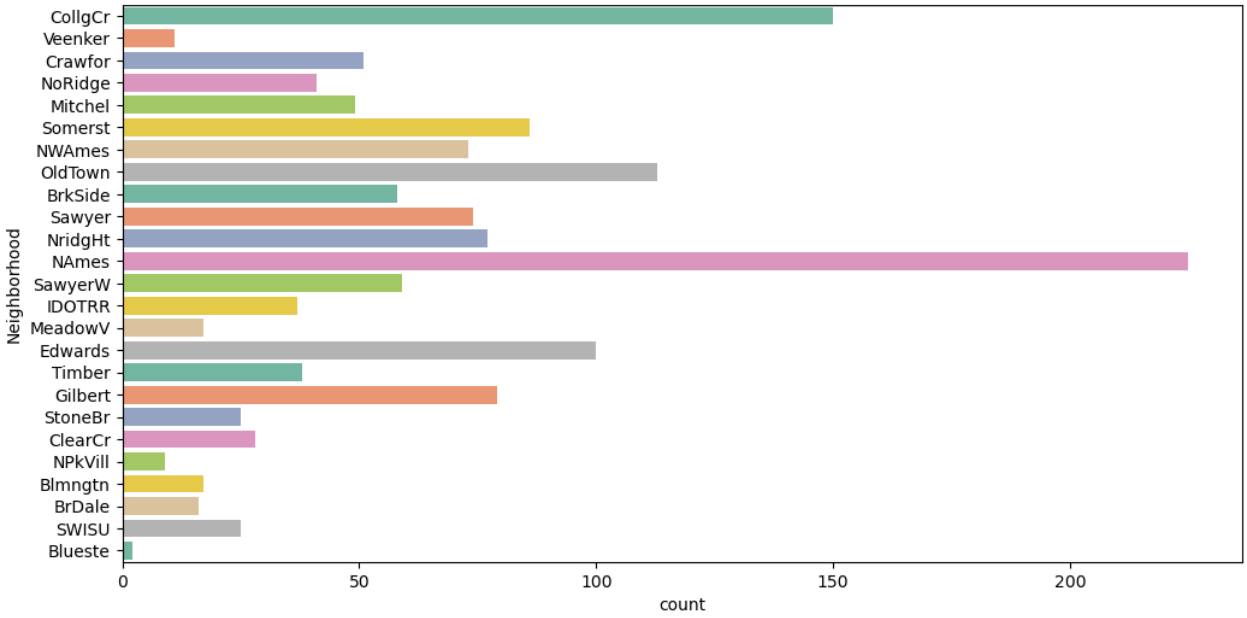


Comment -

1. There are more sales in the Residential areas with low densities(RL)
2. Detached Single-Family houses are most popular among the buyers.
3. Houses with concrete and block foundations have been trusted by buyers.
4. Buyers are very keen with the attached garage to the house.

In [27]:
```
plt.figure(figsize=(12,6))
sns.countplot(y=df['Neighborhood'],palette='Set2')
```

Out[27]:
```
<AxesSubplot:xlabel='count', ylabel='Neighborhood'>
```



It seems like North Ames is the most popular location choice for the home buyers

In [ ]:
```
plt.figure(figsize=(10,6))
sns.boxplot(x='MSSubClass',y='SalePrice',data=df)
```

SC60(2-STORY 1946 & NEWER), SC20(1-STORY 1946 & NEWER ALL STYLES) have seen the highest sales as well as highest Selling prices. And the houses have been sold with lot of variations in Saleprices.

In [ ]:
```
plt.figure(figsize=(10,6))
sns.boxplot(x='SaleType',y='SalePrice',data=df)
```

In [28]:
```
crosstab=pd.crosstab(index=df['Neighborhood'],columns=df['SaleCondition'])
crosstab
```

Out[28]:

Neighborhood \ SaleCondition	Abnorml	AdjLand	Alloca	Family	Normal	Partial
Blmngtn	0	0	0	0	12	5
Blueste	0	0	0	0	2	0
BrDale	3	0	0	1	12	0
BrkSide	3	0	0	1	54	0
ClearCr	3	0	0	0	24	1
CollgCr	3	0	0	0	129	18
Crawfor	3	0	2	2	43	1
Edwards	8	4	2	0	82	4
Gilbert	1	0	0	2	64	12
IDOTRR	7	0	1	0	29	0
MeadowV	1	0	0	0	16	0
Mitchel	3	0	1	2	42	1
NAmes	23	0	0	4	198	0
NPkVill	1	0	0	0	8	0
NWAmes	6	0	0	3	64	0
NoRidge	4	0	0	0	37	0
NridgHt	0	0	0	0	45	32
OldTown	12	0	1	4	94	2
SWISU	3	0	0	0	22	0
Sawyer	5	0	1	1	67	0
SawyerW	4	0	4	0	50	1
Somerst	4	0	0	0	49	33
StoneBr	1	0	0	0	16	8
Timber	3	0	0	0	28	7
Veenker	0	0	0	0	11	0

In [ ]:
```
plt.figure(figsize=(12,12))
sns.heatmap(df.corr(),cmap='viridis' )
```

In [ ]:
```
k = 10   #number of variables for heatmap
columns = df.corr().nlargest(k, 'SalePrice')['SalePrice'].index
CR = df[columns].corr()
plt.figure(figsize=(10,10))
sns.heatmap(CR, annot=True, cmap = 'viridis')
```

In [39]:

```
'GarageQual', 'GarageCond', 'PavedDrive', 'SaleType', 'SaleCondition','SalePrice']
```

```
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 IMPORTANT for Prediction
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 NOT an important predictor. (Discard Condition2 from model)
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 IMPORTANT for Prediction
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 IMPORTANT for Prediction
HeatingQC is NOT an important predictor. (Discard HeatingQC from model)
CentralAir is IMPORTANT for Prediction
Electrical is NOT an important predictor. (Discard Electrical from model)
KitchenQual is IMPORTANT for Prediction
Functional is NOT an important predictor. (Discard Functional from model)
GarageType is NOT an important predictor. (Discard GarageType from model)
GarageFinish is IMPORTANT for Prediction
GarageQual is NOT an important predictor. (Discard GarageQual from model)
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
```

5. Combining all the significant categorical and numerical variables which are stated as important for predictions

In [46]:
```
df1=df[['Street', 'LotShape','LotConfig',
'Neighborhood','MasVnrType', 'ExterQual', 'ExterCond', 'Foundation', 'BsmtQual', 'B
```

In [47]:
```
df1.head(5)
```

Out[47]:

	Street	LotShape	LotConfig	Neighborhood	MasVnrType	ExterQual	ExterCond	Foundation	Bsmt
0	Pave	Reg	Inside	CollgCr	BrkFace	Gd	TA	PConc	
1	Pave	Reg	FR2	Veenker	None	TA	TA	CBlock	
2	Pave	IR1	Inside	CollgCr	BrkFace	Gd	TA	PConc	
3	Pave	IR1	Corner	Crawfor	None	TA	TA	BrkTil	
4	Pave	IR1	FR2	NoRidge	BrkFace	Gd	TA	PConc	

In [45]:
```
df_numerical.head(5)
```

Out[45]:

	Unnamed: 0	LotFrontage	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea
0	0	65	8450	7	5	2003	2003	196
1	1	80	9600	6	8	1976	1976	0
2	2	68	11250	7	5	2001	2002	162
3	3	60	9550	7	5	1915	1970	0
4	4	84	14260	8	5	2000	2000	350

5 rows × 36 columns

In [50]:
```
new_df=pd.merge(df1,df_numerical,how='outer',on=['SalePrice'])
```

In [52]:
```
new_df.head(5)
```

Out[52]:

eCars	GarageArea	WoodDeckSF	OpenPorchSF	EnclosedPorch	3SsnPorch	ScreenPorch	PoolArea	M
2	548	0	61	0	0	0	0	
2	460	298	0	0	0	0	0	
2	608	0	42	0	0	0	0	
2	528	0	312	0	0	0	0	
2	608	0	42	0	0	0	0	

In [54]:
```
new_df.shape
```

Out[54]: (6770, 53)

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

In [55]:
```
#Function to plot all independent categorical variables with SalePrice and count pl
ix = 1
fig = plt.figure(figsize = (15,10))
for c in list(new_df.columns):
    if ix <= 3:
        if c != 'SalePrice':
            ax2 = fig.add_subplot(2,3,ix+3)
            sns.boxplot(data=new_df, x=c, y='SalePrice', ax=ax2) #for boxplot

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