Housing Prices Univariate Analyses

In [1]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as ph
```

In [2]:
```
df=pd.read_csv("D:\\datasets\\train.csv")
df
```

Out[2]:

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

1460 rows × 81 columns

In [3]:
```
def import_housing_data(url):
    import pandas as pd
    df= pd.read_csv(url)
```

```
    df.drop(columns=['Id'], inplace=True)
    return df
df=import_housing_data("D:\\datasets\\train.csv")
df.head()
```

Out[3]:

	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	LotConfig	...	PoolArea	PoolQC	Fence	Mis
0	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	Inside	...	0	NaN	NaN	
1	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	FR2	...	0	NaN	NaN	
2	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	Inside	...	0	NaN	NaN	
3	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	Corner	...	0	NaN	NaN	
4	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	FR2	...	0	NaN	NaN	

5 rows × 80 columns

In [4]:

```
def unistats(df):
    import pandas as pd
    output_df=pd.DataFrame(columns=['Count', 'Missing', 'Unique', 'Dtype' ,'Numeric', 'Mode', 'Mean', 'Min', '25%', 'Medi

    for col in df:
        if pd.api.types.is_numeric_dtype(df[col]):
            output_df.loc[col]= [df[col].count(), df[col].isnull().sum(), df[col].nunique(),
                                 df[col].dtype, pd.api.types.is_numeric_dtype(df[col]),
                                 df[col].mode().values[0], df[col].mean(), df[col].min(),
                                 df[col].quantile(0.25), df[col].median(),
                                df[col].quantile(0.75),df[col].max(), df[col].std(),
                                 df[col].skew(), df[col].kurt()]
        else:
            output_df.loc[col]= [df[col].count(), df[col].isnull().sum(), df[col].nunique(),
                                 df[col].dtype, pd.api.types.is_numeric_dtype(df[col]),
                                 df[col].mode().values[0], '-', '-', '-', '-', '-', '-', '-', '-', '-']
    return output_df.sort_values(by=['Numeric','Skew', 'Unique'], ascending=False)

#Test The Fyncrion
import pandas as pd
pd.set_option('display.max_rows', 200)
pd.set_option('display.max_columns', 200)
```

```
df=pd.read_csv("D:\\datasets\\train.csv")

unistats(df)
```

Out[4]:

	Count	Missing	Unique	Dtype	Numeric	Mode	Mean	Min	25%	Median	75%	Max	Std
MiscVal	1460	0	21	int64	True	0	43.489041	0	0.0	0.0	0.0	15500	496.123024
PoolArea	1460	0	8	int64	True	0	2.758904	0	0.0	0.0	0.0	738	40.177307
LotArea	1460	0	1073	int64	True	7200	10516.828082	1300	7553.5	9478.5	11601.5	215245	9981.264932
3SsnPorch	1460	0	20	int64	True	0	3.409589	0	0.0	0.0	0.0	508	29.317331
LowQualFinSF	1460	0	24	int64	True	0	5.844521	0	0.0	0.0	0.0	572	48.623081
KitchenAbvGr	1460	0	4	int64	True	1	1.046575	0	1.0	1.0	1.0	3	0.220338
BsmtFinSF2	1460	0	144	int64	True	0	46.549315	0	0.0	0.0	0.0	1474	161.319273
ScreenPorch	1460	0	76	int64	True	0	15.060959	0	0.0	0.0	0.0	480	55.757415
BsmtHalfBath	1460	0	3	int64	True	0	0.057534	0	0.0	0.0	0.0	2	0.238753
EnclosedPorch	1460	0	120	int64	True	0	21.95411	0	0.0	0.0	0.0	552	61.119149
MasVnrArea	1452	8	327	float64	True	0.0	103.685262	0.0	0.0	0.0	166.0	1600.0	181.066207
OpenPorchSF	1460	0	202	int64	True	0	46.660274	0	0.0	25.0	68.0	547	66.256028
LotFrontage	1201	259	110	float64	True	60.0	70.049958	21.0	59.0	69.0	80.0	313.0	24.284752
SalePrice	1460	0	663	int64	True	140000	180921.19589	34900	129975.0	163000.0	214000.0	755000	79442.502883
BsmtFinSF1	1460	0	637	int64	True	0	443.639726	0	0.0	383.5	712.25	5644	456.098091
WoodDeckSF	1460	0	274	int64	True	0	94.244521	0	0.0	0.0	168.0	857	125.338794
TotalBsmtSF	1460	0	721	int64	True	0	1057.429452	0	795.75	991.5	1298.25	6110	438.705324
MSSubClass	1460	0	15	int64	True	20	56.89726	20	20.0	50.0	70.0	190	42.300571
1stFlrSF	1460	0	753	int64	True	864	1162.626712	334	882.0	1087.0	1391.25	4692	386.587738
GrLivArea	1460	0	861	int64	True	864	1515.463699	334	1129.5	1464.0	1776.75	5642	525.480383
BsmtUnfSF	1460	0	780	int64	True	0	567.240411	0	223.0	477.5	808.0	2336	441.866955
2ndFlrSF	1460	0	417	int64	True	0	346.992466	0	0.0	0.0	728.0	2065	436.528436
OverallCond	1460	0	9	int64	True	5	5.575342	1	5.0	5.0	6.0	9	1.112799

	Count	Missing	Unique	Dtype	Numeric	Mode	Mean	Min	25%	Median	75%	Max	Std
TotRmsAbvGrd	1460	0	12	int64	True	6	6.517808	2	5.0	6.0	7.0	14	1.625393
HalfBath	1460	0	3	int64	True	0	0.382877	0	0.0	0.0	1.0	2	0.502885
Fireplaces	1460	0	4	int64	True	0	0.613014	0	0.0	1.0	1.0	3	0.644666
BsmtFullBath	1460	0	4	int64	True	0	0.425342	0	0.0	0.0	1.0	3	0.518911
OverallQual	1460	0	10	int64	True	5	6.099315	1	5.0	6.0	7.0	10	1.382997
MoSold	1460	0	12	int64	True	6	6.321918	1	5.0	6.0	8.0	12	2.703626
BedroomAbvGr	1460	0	8	int64	True	3	2.866438	0	2.0	3.0	3.0	8	0.815778
GarageArea	1460	0	441	int64	True	0	472.980137	0	334.5	480.0	576.0	1418	213.804841
YrSold	1460	0	5	int64	True	2009	2007.815753	2006	2007.0	2008.0	2009.0	2010	1.328095
FullBath	1460	0	4	int64	True	2	1.565068	0	1.0	2.0	2.0	3	0.550916
Id	1460	0	1460	int64	True	1	730.5	1	365.75	730.5	1095.25	1460	421.610009
GarageCars	1460	0	5	int64	True	2	1.767123	0	1.0	2.0	2.0	4	0.747315
YearRemodAdd	1460	0	61	int64	True	1950	1984.865753	1950	1967.0	1994.0	2004.0	2010	20.645407
YearBuilt	1460	0	112	int64	True	2006	1971.267808	1872	1954.0	1973.0	2000.0	2010	30.202904
GarageYrBlt	1379	81	97	float64	True	2005.0	1978.506164	1900.0	1961.0	1980.0	2002.0	2010.0	24.689725
Neighborhood	1460	0	25	object	False	NAmes	-	-	-	-	-	-	-
Exterior2nd	1460	0	16	object	False	VinylSd	-	-	-	-	-	-	-
Exterior1st	1460	0	15	object	False	VinylSd	-	-	-	-	-	-	-
Condition1	1460	0	9	object	False	Norm	-	-	-	-	-	-	-
SaleType	1460	0	9	object	False	WD	-	-	-	-	-	-	-
Condition2	1460	0	8	object	False	Norm	-	-	-	-	-	-	-
HouseStyle	1460	0	8	object	False	1Story	-	-	-	-	-	-	-
RoofMatl	1460	0	8	object	False	CompShg	-	-	-	-	-	-	-
Functional	1460	0	7	object	False	Typ	-	-	-	-	-	-	-
RoofStyle	1460	0	6	object	False	Gable	-	-	-	-	-	-	-

	Count	Missing	Unique	Dtype	Numeric	Mode	Mean	Min	25%	Median	75%	Max	Std
Foundation	1460	0	6	object	False	PConc	-	-	-	-	-	-	-
BsmtFinType1	1423	37	6	object	False	Unf	-	-	-	-	-	-	-
BsmtFinType2	1422	38	6	object	False	Unf	-	-	-	-	-	-	-
Heating	1460	0	6	object	False	GasA	-	-	-	-	-	-	-
GarageType	1379	81	6	object	False	Attchd	-	-	-	-	-	-	-
SaleCondition	1460	0	6	object	False	Normal	-	-	-	-	-	-	-
MSZoning	1460	0	5	object	False	RL	-	-	-	-	-	-	-
LotConfig	1460	0	5	object	False	Inside	-	-	-	-	-	-	-
BldgType	1460	0	5	object	False	1Fam	-	-	-	-	-	-	-
ExterCond	1460	0	5	object	False	TA	-	-	-	-	-	-	-
HeatingQC	1460	0	5	object	False	Ex	-	-	-	-	-	-	-
Electrical	1459	1	5	object	False	SBrkr	-	-	-	-	-	-	-
FireplaceQu	770	690	5	object	False	Gd	-	-	-	-	-	-	-
GarageQual	1379	81	5	object	False	TA	-	-	-	-	-	-	-
GarageCond	1379	81	5	object	False	TA	-	-	-	-	-	-	-
LotShape	1460	0	4	object	False	Reg	-	-	-	-	-	-	-
LandContour	1460	0	4	object	False	Lvl	-	-	-	-	-	-	-
MasVnrType	1452	8	4	object	False	None	-	-	-	-	-	-	-
ExterQual	1460	0	4	object	False	TA	-	-	-	-	-	-	-
BsmtQual	1423	37	4	object	False	TA	-	-	-	-	-	-	-
BsmtCond	1423	37	4	object	False	TA	-	-	-	-	-	-	-
BsmtExposure	1422	38	4	object	False	No	-	-	-	-	-	-	-
KitchenQual	1460	0	4	object	False	TA	-	-	-	-	-	-	-
Fence	281	1179	4	object	False	MnPrv	-	-	-	-	-	-	-
MiscFeature	54	1406	4	object	False	Shed	-	-	-	-	-	-	-

	Count	Missing	Unique	Dtype	Numeric	Mode	Mean	Min	25%	Median	75%	Max	Std
LandSlope	1460	0	3	object	False	Gtl	-	-	-	-	-	-	-
GarageFinish	1379	81	3	object	False	Unf	-	-	-	-	-	-	-
PavedDrive	1460	0	3	object	False	Y	-	-	-	-	-	-	-
PoolQC	7	1453	3	object	False	Gd	-	-	-	-	-	-	-
Street	1460	0	2	object	False	Pave	-	-	-	-	-	-	-
Alley	91	1369	2	object	False	Grvl	-	-	-	-	-	-	-
Utilities	1460	0	2	object	False	AllPub	-	-	-	-	-	-	-
CentralAir	1460	0	2	object	False	Y	-	-	-	-	-	-	-

Housing Prices Bivariate Statistics

In [5]:
```
def anvoa(df, feature, label):
    from scipy import stats
    import pandas as pd
    import numpy as np

    groups= df[feature].unique()
    df_grouped= df.groupby(feature)
    group_label
    for g in groups:
        g_list= df.grouped.get_group(g)
        group_labels.append(g_list[label])

        return stats.f_oneway(*group_labels)
```

In [6]:
```
#Bivariate:Numeric To Numeric: Correlation
#Bivariate:Numeric To Categorical: One_way ANOVA (3+ groups) or t-test (2 groups)
#Bivariate:Categorical To Categorical:Chi-square

def bivstats(df, label):
    from scipy import stats
    import pandas as pd
```

```
    import numpy as np

    #Creat an empty DataFrame to store output
    output_df=pd.DataFrame(columns=['r', 'F', 'x2', 'p-value'])

    for col in df:
        if not col == label:
            if df[col].isnull().sum() ==0:
                 if pd.api.types.is_numeric_dtype(df[col]): #only calculate r , p-value for the numeric columns
                    r, p = stats.pearsonr(df[label], df[col])
                    output_df.loc[col]=[round(r, 3), np.nan, np.nan, round(p, 3)]
        else:
            output_df.loc[col]= [np.nan, np.nan, np.nan, 'nulls']

    return output_df.reindex(output_df.r.abs().sort_values(ascending=False).index)

import pandas as pd
df=pd.read_csv("D:\\datasets\\train.csv")
bivstats(df, 'SalePrice')
```

Out[6]:

	r	F	x2	p-value
OverallQual	0.791	NaN	NaN	0.0
GrLivArea	0.709	NaN	NaN	0.0
GarageCars	0.640	NaN	NaN	0.0
GarageArea	0.623	NaN	NaN	0.0
TotalBsmtSF	0.614	NaN	NaN	0.0
1stFlrSF	0.606	NaN	NaN	0.0
FullBath	0.561	NaN	NaN	0.0
TotRmsAbvGrd	0.534	NaN	NaN	0.0
YearBuilt	0.523	NaN	NaN	0.0
YearRemodAdd	0.507	NaN	NaN	0.0
Fireplaces	0.467	NaN	NaN	0.0
BsmtFinSF1	0.386	NaN	NaN	0.0
WoodDeckSF	0.324	NaN	NaN	0.0

	r	F	x2	p-value
2ndFlrSF	0.319	NaN	NaN	0.0
OpenPorchSF	0.316	NaN	NaN	0.0
HalfBath	0.284	NaN	NaN	0.0
LotArea	0.264	NaN	NaN	0.0
BsmtFullBath	0.227	NaN	NaN	0.0
BsmtUnfSF	0.214	NaN	NaN	0.0
BedroomAbvGr	0.168	NaN	NaN	0.0
KitchenAbvGr	-0.136	NaN	NaN	0.0
EnclosedPorch	-0.129	NaN	NaN	0.0
ScreenPorch	0.111	NaN	NaN	0.0
PoolArea	0.092	NaN	NaN	0.0
MSSubClass	-0.084	NaN	NaN	0.001
OverallCond	-0.078	NaN	NaN	0.003
MoSold	0.046	NaN	NaN	0.076
3SsnPorch	0.045	NaN	NaN	0.089
YrSold	-0.029	NaN	NaN	0.269
LowQualFinSF	-0.026	NaN	NaN	0.328
Id	-0.022	NaN	NaN	0.403
MiscVal	-0.021	NaN	NaN	0.418
BsmtHalfBath	-0.017	NaN	NaN	0.52
BsmtFinSF2	-0.011	NaN	NaN	0.664
SalePrice	NaN	NaN	NaN	nulls

In [7]:
```
#Bivariate:Numeric To Numeric: Correlation
#Bivariate:Numeric To Categorical: One_way ANOVA (3+ groups) or t-test (2 groups)
#Bivariate:Categorical To Categorical:Chi-square
```

```
def bivstats(df, label):
    from scipy import stats
    import pandas as pd
    import numpy as np

    #Creat an empty DataFrame to store output
    output_df=pd.DataFrame(columns=['stat', '+/-', 'Effect size', 'p-value'])

    for col in df:
        if not col == label:
            if df[col].isnull().sum() ==0:
                if pd.api.types.is_numeric_dtype(df[col]): #only calculate r , p-value for the numeric columns
                    r, p = stats.pearsonr(df[label], df[col])
                    output_df.loc[col]=['r', np.sign(r), abs(round(r, 3)), round(p, 6)]
                else:
                    F, p = anova(df[[col, label]], col, label)
                    output_df.loc[col]=['F', '', round(F, 3), round(p, 6)]
            else:
                output_df.loc[col]= [np.nan, np.nan, np.nan, np.nan]

    #return output_df.reindex(output_df.r.abs().sort_values(ascending=False).index)
    return output_df.sort_values(by=['Effect size', 'stat'], ascending=[False, False])

import pandas as pd
df=pd.read_csv("D:\\datasets\\train.csv")
bivstats(df, 'SalePrice')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
C:\Users\TAWABC~1\AppData\Local\Temp/ipykernel_10864/3705386330.py in <module>
     28 import pandas as pd
     29 df=pd.read_csv("D:\\datasets\\train.csv")
---> 30 bivstats(df, 'SalePrice')

C:\Users\TAWABC~1\AppData\Local\Temp/ipykernel_10864/3705386330.py in bivstats(df, label)
     18                     output_df.loc[col]=['r', np.sign(r), abs(round(r, 3)), round(p, 6)]
     19                 else:
---> 20                     F, p = anova(df[[col, label]], col, label)
     21                     output_df.loc[col]=['F', '', round(F, 3), round(p, 6)]
     22             else:

NameError: name 'anova' is not defined
```

In [8]:

```
def bivstats(df, label):
    from scipy import stats
    import pandas as pd
    import numpy as np

    #Creat an empty DataFrame to store output
    output_df=pd.DataFrame(columns=['stat', '+/-', 'Effect size', 'p-value'])

    for col in df:
        if not col == label:
            if df[col].isnull().sum() ==0:
                if pd.api.types.is_numeric_dtype(df[col]): #only calculate r , p-value for the numeric columns
                    r, p = stats.pearsonr(df[label], df[col])
                    output_df.loc[col]=['r', np.sign(r), abs(round(r, 3)), round(p, 6)]
                else:
                    F, p = anova(df[[col, label]], col, label)
                    output_df.loc[col]=['F', '', round(F, 3), round(p, 6)]
            else:
                output_df.loc[col]= [np.nan, np.nan, np.nan, np.nan]

    #return output_df.reindex(output_df.r.abs().sort_values(ascending=False).index)
    return output_df.sort_values(by=['Effect size', 'stat'], ascending=[False, False])

import pandas as pd
df=pd.read_csv("D:\\datasets\\train.csv")
bivstats(df, 'SalePrice')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
C:\Users\TAWABC~1\AppData\Local\Temp/ipykernel_10864/3386843493.py in <module>
     24 import pandas as pd
     25 df=pd.read_csv("D:\\datasets\\train.csv")
---> 26 bivstats(df, 'SalePrice')

C:\Users\TAWABC~1\AppData\Local\Temp/ipykernel_10864/3386843493.py in bivstats(df, label)
     14                     output_df.loc[col]=['r', np.sign(r), abs(round(r, 3)), round(p, 6)]
     15                 else:
---> 16                     F, p = anova(df[[col, label]], col, label)
     17                     output_df.loc[col]=['F', '', round(F, 3), round(p, 6)]
     18             else:

NameError: name 'anova' is not defined
```

Housing Prices Bivariate Visualizations

In [ ]: