Assignment 3 - Part 3

In this assignment I will do the research about prices of the houses, how it is disturbed and which caracteristics of the house are affecting in's price. For this purpose I will use this dataset: House Price Dataset.

In [2]:
```
# import data analysis libraries
import pandas as pd

# import data visualization libraries
import seaborn as sns
from matplotlib import pyplot as plt
%matplotlib inline
```

In [3]:
```
house = pd.read_csv("train.csv")     # reading dataset and storing it under name house
```

In [4]:
```
house.head()   # look at the first few rows of the dataset to observe what all we have in it
```

Out[4]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	MiscFeat
0	1	60	RL	65.0	8450	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	N
1	2	20	RL	80.0	9600	Pave	NaN	Reg	Lvl	AllPub	...	0	NaN	NaN	N
2	3	60	RL	68.0	11250	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	N
3	4	70	RL	60.0	9550	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	N
4	5	60	RL	84.0	14260	Pave	NaN	IR1	Lvl	AllPub	...	0	NaN	NaN	N

5 rows × 81 columns

In [5]:
```
house.info()      # info is giving us all information about variables in this dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1460 entries, 0 to 1459
Data columns (total 81 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Id             1460 non-null   int64  
 1   MSSubClass     1460 non-null   int64  
 2   MSZoning       1460 non-null   object 
 3   LotFrontage    1201 non-null   float64
 4   LotArea        1460 non-null   int64  
 5   Street         1460 non-null   object 
 6   Alley          91 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     1452 non-null   object 
 26  MasVnrArea     1452 non-null   float64
 27  ExterQual      1460 non-null   object 
 28  ExterCond      1460 non-null   object 
 29  Foundation     1460 non-null   object 
 30  BsmtQual       1423 non-null   object 
 31  BsmtCond       1423 non-null   object 
 32  BsmtExposure   1422 non-null   object 
 33  BsmtFinType1   1423 non-null   object 
 34  BsmtFinSF1     1460 non-null   int64  
 35  BsmtFinType2   1422 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    770 non-null    object 
 58  GarageType     1379 non-null   object 
 59  GarageYrBlt    1379 non-null   float64
 60  GarageFinish   1379 non-null   object 
 61  GarageCars     1460 non-null   int64  
 62  GarageArea     1460 non-null   int64  
 63  GarageQual     1379 non-null   object 
 64  GarageCond     1379 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         7 non-null      object 
 73  Fence          281 non-null    object 
 74  MiscFeature    54 non-null     object 
 75  MiscVal        1460 non-null   int64  
 76  MoSold         1460 non-null   int64  
 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(3), int64(35), object(43)
memory usage: 924.0+ KB
```

First thing that we do after looking at data types in data set is to deal with null values or missing values. In this research I will just delete columns which have very big amount of missing values, because that kind of data will not be usefull and it is hard to manage on any other way like filling or deleting observations because huge number will be lost. And also these columns are not affecting on Saleprice very much and we have other simmilar caracteristics that have better correlation as we will see soon.

In [6]:
```
house.isnull()   #looking at data and checking for null-s
```

Out[6]:

	Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContour	Utilities	...	PoolArea	PoolQC	Fence	M
0	False	False	False	False	False	False	True	False	False	False	...	False	True	True	
1	False	False	False	False	False	False	True	False	False	False	...	False	True	True	
2	False	False	False	False	False	False	True	False	False	False	...	False	True	True	
3	False	False	False	False	False	False	True	False	False	False	...	False	True	True	
4	False	False	False	False	False	False	True	False	False	False	...	False	True	True	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
1455	False	False	False	False	False	False	True	False	False	False	...	False	True	True	
1456	False	False	False	False	False	False	True	False	False	False	...	False	True	False	
1457	False	False	False	False	False	False	True	False	False	False	...	False	True	False	
1458	False	False	False	False	False	False	True	False	False	False	...	False	True	True	
1459	False	False	False	False	False	False	True	False	False	False	...	False	True	True	

1460 rows × 81 columns

We can immidiatly see that there are some columns with a lot of null values. Let's see which columns have the most missing data

In [7]:
```
house.isnull().sum()  # let's see how much missing values we have in each column first
```

Out[7]:
```
Id                 0
MSSubClass         0
MSZoning           0
LotFrontage      259
LotArea            0
                ... 
MoSold             0
YrSold             0
SaleType           0
SaleCondition      0
SalePrice          0
Length: 81, dtype: int64
```

In [8]:
```
nul = house.isnull().sum()/1460 * 100  # now let's see in percentage which columns has higher number(percent) o
nul.sort_values(ascending = False)
```

Out[8]:
```
PoolQC          99.520548
MiscFeature     96.301370
Alley           93.767123
Fence           80.753425
FireplaceQu     47.260274
                  ...    
ExterQual        0.000000
Exterior2nd      0.000000
Exterior1st      0.000000
RoofMatl         0.000000
SalePrice        0.000000
Length: 81, dtype: float64
```

Here we can see that there are many variables that have more then 15% of missing data. We will remove whole columns with so many missing data.

In [9]:
```
house1 = house

for x in house.columns:
    if (house1[x].isnull().sum()/1460*100>15):
        house1.drop(x, axis=1, inplace=True)
house1
```

Out[9]:

	Id	MSSubClass	MSZoning	LotArea	Street	LotShape	LandContour	Utilities	LotConfig	LandSlope	...	EnclosedPorch	3SsnPorch
0	1	60	RL	8450	Pave	Reg	Lvl	AllPub	Inside	Gtl	...	0	0
1	2	20	RL	9600	Pave	Reg	Lvl	AllPub	FR2	Gtl	...	0	0
2	3	60	RL	11250	Pave	IR1	Lvl	AllPub	Inside	Gtl	...	0	0
3	4	70	RL	9550	Pave	IR1	Lvl	AllPub	Corner	Gtl	...	272	0
4	5	60	RL	14260	Pave	IR1	Lvl	AllPub	FR2	Gtl	...	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
1455	1456	60	RL	7917	Pave	Reg	Lvl	AllPub	Inside	Gtl	...	0	0
1456	1457	20	RL	13175	Pave	Reg	Lvl	AllPub	Inside	Gtl	...	0	0
1457	1458	70	RL	9042	Pave	Reg	Lvl	AllPub	Inside	Gtl	...	0	0
1458	1459	20	RL	9717	Pave	Reg	Lvl	AllPub	Inside	Gtl	...	112	0
1459	1460	20	RL	9937	Pave	Reg	Lvl	AllPub	Inside	Gtl	...	0	0

1460 rows × 75 columns

Now is time to calculate the correlation between variables. This dataset has a lot of variables, so we will use corr function on whole data set to calculate all possible correlations among variables.

In [10]:

```
house1.corr()     # calculating correlation for all dataset, we get matrix of all possible correlation values b
```

Out[10]:

	Id	MSSubClass	LotArea	OverallQual	OverallCond	YearBuilt	YearRemodAdd	MasVnrArea	BsmtFinSF1	BsmtFinS
Id	1.000000	0.011156	-0.033226	-0.028365	0.012609	-0.012713	-0.021998	-0.050298	-0.005024	-0.0059
MSSubClass	0.011156	1.000000	-0.139781	0.032628	-0.059316	0.027850	0.040581	0.022936	-0.069836	-0.0656
LotArea	-0.033226	-0.139781	1.000000	0.105806	-0.005636	0.014228	0.013788	0.104160	0.214103	0.1111
OverallQual	-0.028365	0.032628	0.105806	1.000000	-0.091932	0.572323	0.550684	0.411876	0.239666	-0.0591
OverallCond	0.012609	-0.059316	-0.005636	-0.091932	1.000000	-0.375983	0.073741	-0.128101	-0.046231	0.0402
YearBuilt	-0.012713	0.027850	0.014228	0.572323	-0.375983	1.000000	0.592855	0.315707	0.249503	-0.0491
YearRemodAdd	-0.021998	0.040581	0.013788	0.550684	0.073741	0.592855	1.000000	0.179618	0.128451	-0.0677
MasVnrArea	-0.050298	0.022936	0.104160	0.411876	-0.128101	0.315707	0.179618	1.000000	0.264736	-0.0723
BsmtFinSF1	-0.005024	-0.069836	0.214103	0.239666	-0.046231	0.249503	0.128451	0.264736	1.000000	-0.0501
BsmtFinSF2	-0.005968	-0.065649	0.111170	-0.059119	0.040229	-0.049107	-0.067759	-0.072319	-0.050117	1.0000
BsmtUnfSF	-0.007940	-0.140759	-0.002618	0.308159	-0.136841	0.149040	0.181133	0.114442	-0.495251	-0.2092
TotalBsmtSF	-0.015415	-0.238518	0.260833	0.537808	-0.171098	0.391452	0.291066	0.363936	0.522396	0.1048
1stFlrSF	0.010496	-0.251758	0.299475	0.476224	-0.144203	0.281986	0.240379	0.344501	0.445863	0.0971
2ndFlrSF	0.005590	0.307886	0.050986	0.295493	0.028942	0.010308	0.140024	0.174561	-0.137079	-0.0992
LowQualFinSF	-0.044230	0.046474	0.004779	-0.030429	0.025494	-0.183784	-0.062419	-0.069071	-0.064503	0.0148
GrLivArea	0.008273	0.074853	0.263116	0.593007	-0.079686	0.199010	0.287389	0.390857	0.208171	-0.0096
BsmtFullBath	0.002289	0.003491	0.158155	0.111098	-0.054942	0.187599	0.119470	0.085310	0.649212	0.1586
BsmtHalfBath	-0.020155	-0.002333	0.048046	-0.040150	0.117821	-0.038162	-0.012337	0.026673	0.067418	0.0709
FullBath	0.005587	0.131608	0.126031	0.550600	-0.194149	0.468271	0.439046	0.276833	0.058543	-0.0764
HalfBath	0.006784	0.177354	0.014259	0.273458	-0.060769	0.242656	0.183331	0.201444	0.004262	-0.0321
BedroomAbvGr	0.037719	-0.023438	0.119690	0.101676	0.012980	-0.070651	-0.040581	0.102821	-0.107355	-0.0157
KitchenAbvGr	0.002951	0.281721	-0.017784	-0.183882	-0.087001	-0.174800	-0.149598	-0.037610	-0.081007	-0.0407
TotRmsAbvGrd	0.027239	0.040380	0.190015	0.427452	-0.057583	0.095589	0.191740	0.280682	0.044316	-0.0352
Fireplaces	-0.019772	-0.045569	0.271364	0.396765	-0.023820	0.147716	0.112581	0.249070	0.260011	0.0469
GarageYrBlt	0.000072	0.085072	-0.024947	0.547766	-0.324297	0.825667	0.642277	0.252691	0.153484	-0.0880
GarageCars	0.016570	-0.040110	0.154871	0.600671	-0.185758	0.537850	0.420622	0.364204	0.224054	-0.0382
GarageArea	0.017634	-0.098672	0.180403	0.562022	-0.151521	0.478954	0.371600	0.373066	0.296970	-0.0182
WoodDeckSF	-0.029643	-0.012579	0.171698	0.238923	-0.003334	0.224880	0.205726	0.159718	0.204306	0.0678
OpenPorchSF	-0.000477	-0.006100	0.084774	0.308819	-0.032589	0.188686	0.226298	0.125703	0.111761	0.0030
EnclosedPorch	0.002889	-0.012037	-0.018340	-0.113937	0.070356	-0.387268	-0.193919	-0.110204	-0.102303	0.0365
3SsnPorch	-0.046635	-0.043825	0.020423	0.030371	0.025504	0.031355	0.045286	0.018796	0.026451	-0.0299
ScreenPorch	0.001330	-0.026030	0.043160	0.064886	0.054811	-0.050364	-0.038740	0.061466	0.062021	0.0888
PoolArea	0.057044	0.008283	0.077672	0.065166	-0.001985	0.004950	0.005829	0.011723	0.140491	0.0417
MiscVal	-0.006242	-0.007683	0.038068	-0.031406	0.068777	-0.034383	-0.010286	-0.029815	0.003571	0.0049
MoSold	0.021172	-0.013585	0.001205	0.070815	-0.003511	0.012398	0.021490	-0.005965	-0.015727	-0.0152
YrSold	0.000712	-0.021407	-0.014261	-0.027347	0.043950	-0.013618	0.035743	-0.008201	0.014359	0.0317
SalePrice	-0.021917	-0.084284	0.263843	0.790982	-0.077856	0.522897	0.507101	0.477493	0.386420	-0.0113

37 rows × 37 columns

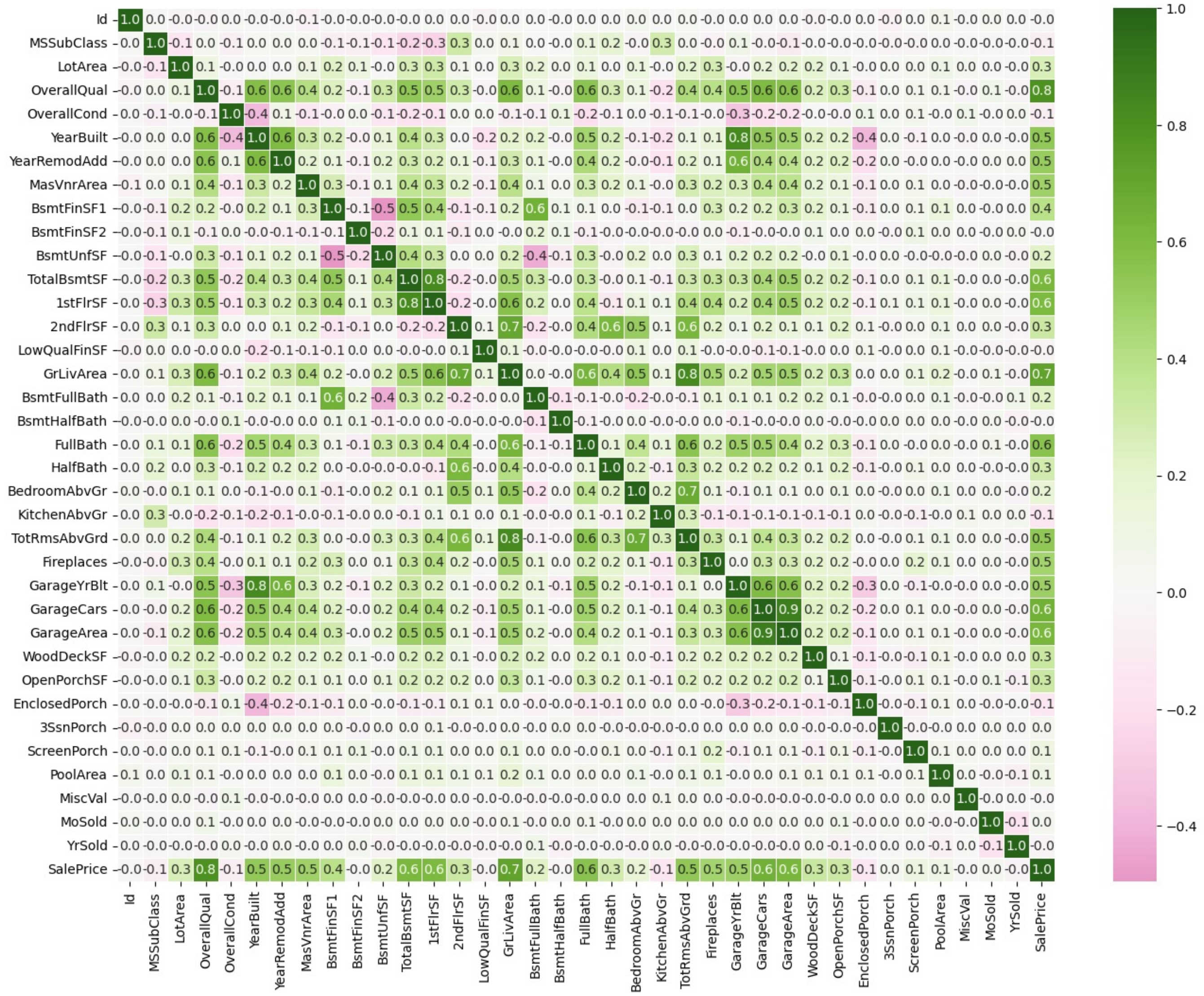
There is one more obvious way to do correlation, using heatmap. From where we can see which variables have the highest correlation and then I will observe relationship between those variables on plots.

In [11]:

```
plt.subplots(figsize=(16, 12))
sns.heatmap(house1.corr(), cmap="PiYG", center=0, annot=True, annot_kws={"size": 10}, fmt='.1f', linewidths=0.5
```

Out[11]:

```
<AxesSubplot:>
```



We can see from heatmap that there is few variables that have correlation between each other. It is not very interesting explaining why there is correlation between YearBuilt and GarageYrBlt, or GrLivArea and TotRmsAbvGrd, or TotRmsAbvGr and BedroomAbvGr, or GarageArea and GarageCars... Those correlations are obvious and they don't give us any interesting conclusions. The most interesting thing that can be observed here is correlation between SalesPrice and other variables. On that way we can see which characteristics of house mostly affect her price.

In [12]:

```
corrSale = house1.corr()['SalePrice']                    # so we are extracting only correlations with SalePri
corrSale = corrSale.abs().sort_values(ascending=False)   # and then sorting them in decreasing order from highe
corrSale
```

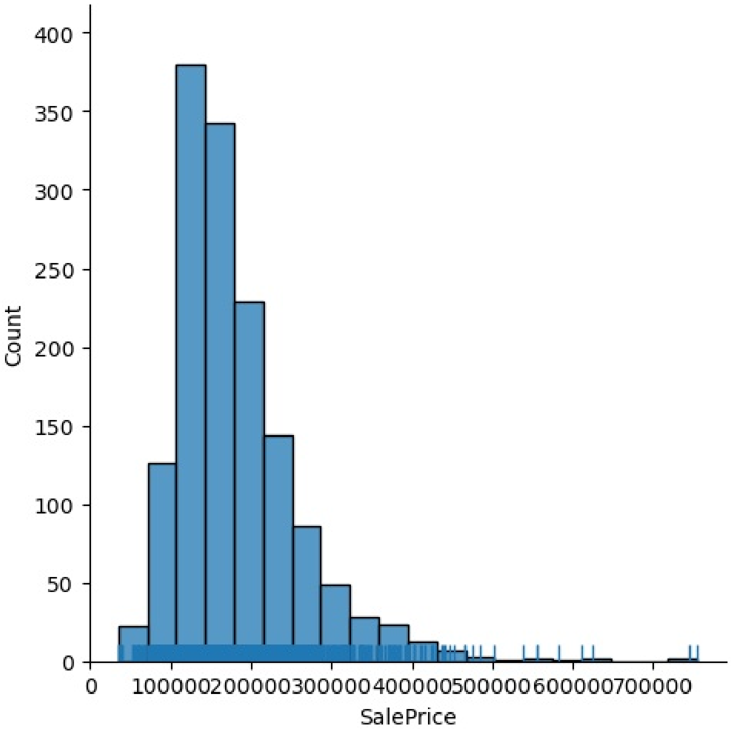
Out[12]:
```
SalePrice        1.000000
OverallQual      0.790982
GrLivArea        0.708624
GarageCars       0.640409
GarageArea       0.623431
TotalBsmtSF      0.613581
1stFlrSF         0.605852
FullBath         0.560664
TotRmsAbvGrd     0.533723
YearBuilt        0.522897
YearRemodAdd     0.507101
GarageYrBlt      0.486362
MasVnrArea       0.477493
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
KitchenAbvGr     0.135907
EnclosedPorch    0.128578
ScreenPorch      0.111447
PoolArea         0.092404
MSSubClass       0.084284
OverallCond      0.077856
MoSold           0.046432
3SsnPorch        0.044584
YrSold           0.028923
LowQualFinSF     0.025606
Id               0.021917
MiscVal          0.021190
BsmtHalfBath     0.016844
BsmtFinSF2       0.011378
Name: SalePrice, dtype: float64
```

Now let's look at SalePrice variable and its distribution. We can calculate mean, standard deviation and other statistical measures.

In [18]:
```
sns.displot(house['SalePrice'],bins=20,kind='hist',rug=True)

house['SalePrice'].describe()
```

Out[18]:
```
count      1460.000000
mean     180921.195890
std       79442.502883
min       34900.000000
25%      129975.000000
50%      163000.000000
75%      214000.000000
max      755000.000000
Name: SalePrice, dtype: float64
```



We can see that it is right skewed and that most houses have price under 300.000, mean value is 180921.195890. Standard deviation is 79442.502883. Minimal price is 34900. and maximal price is 755000. Now let's see which caracteristics affect price of house the most

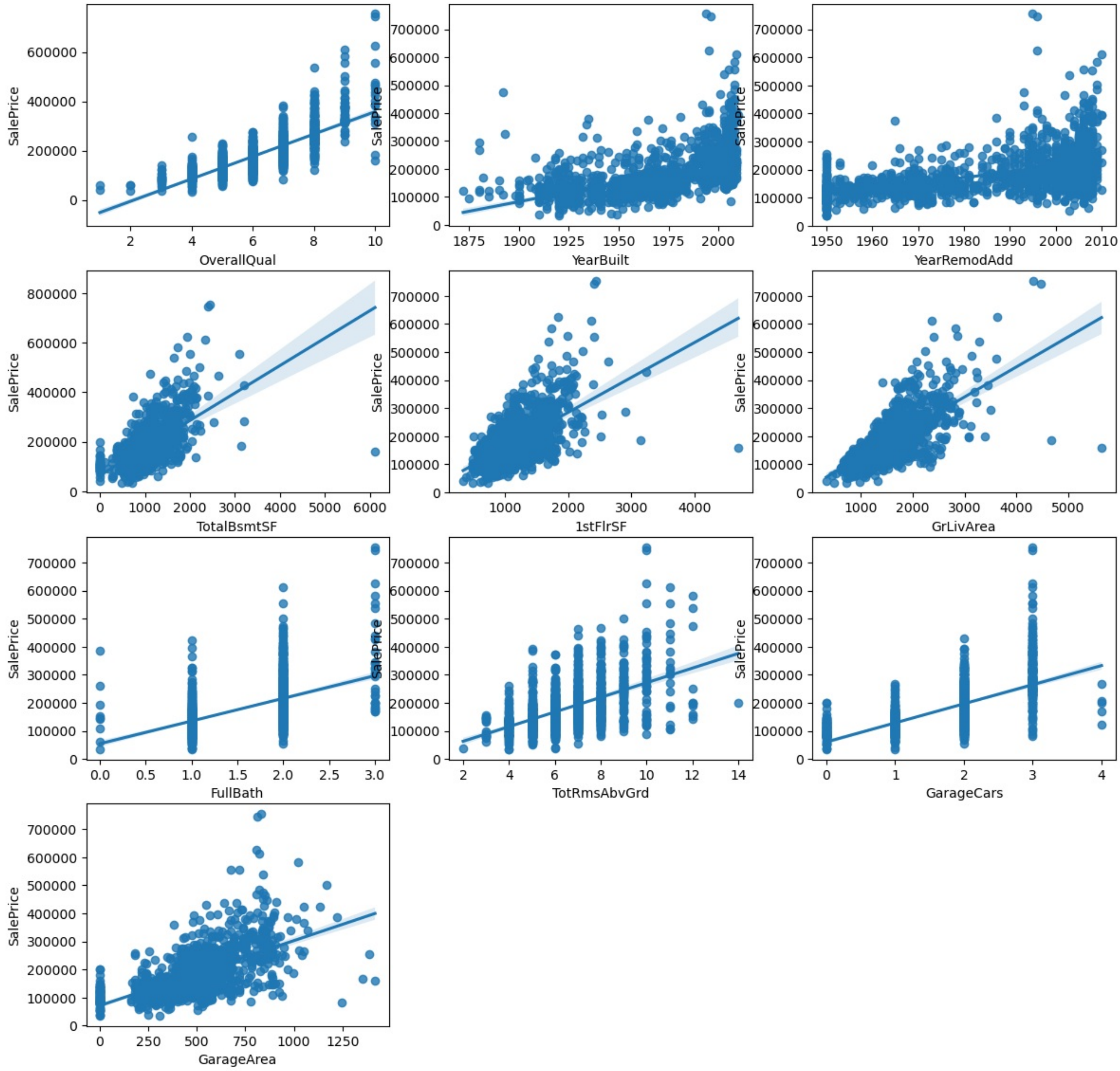
In [20]:
```
important = list(house1.corr()["SalePrice"][(house1.corr()["SalePrice"]>0.50) | (house1.corr()["SalePrice"]<-0.
important.remove('SalePrice')
important
# those variables have strong correlation with SalePrice so I will separate them in one list
```

Out[20]:
```
['OverallQual',
 'YearBuilt',
 'YearRemodAdd',
 'TotalBsmtSF',
 '1stFlrSF',
 'GrLivArea',
 'FullBath',
 'TotRmsAbvGrd',
 'GarageCars',
 'GarageArea']
```

All of this important columns that have high correlation with SalePrice are numerical, so we can look at scaterplots for each to observe what kind of correlation they have. I will use subplot for that, so I can put more plots one by one and observe

In [21]:
```
plt.figure(figsize=(14,14))
i = 0

for xx in important:
    plt.subplot(4,3,i+1)
    sns.regplot(x=house1[xx],y='SalePrice',data=house1)
    i = i+1
```



As we expected there is strong, positive correlation between SalePrice and TotalBsmtSF, 1stFlrSF, GrLivArea, GarageArea, OverallQual, YearBuilt, YearRemodAdd. And also there are some outliers in there. Other things that we can see from plots are:

- TotalBsmtSF, 1stFlrSF, and GrLivArea have very simmilar correlation with HousePrice, which is logical, because those three variables are saying same thing, size of the house and that of course affect it's price, but from this plots we can see that size of house can affect change in it's price very fast, small increase in size makes very big increase in SalePrice.

- GarageArea and GarageCars are also very simmilar and that is because bigger GarageArea means more cars can fit into Garage.
- When we look at the YearBuilt and YearRemodAdd we can say that there is correlation between them and SalePrice, again linear but we can say that year is not affecting on price same as size of house. So we can conclude that new houses are more expencive, but the difference is not that high as it is in other categories.
- OverallQual has the strongest correlation with SalePrice which is also logical, because overall better houses means better in all categories so of course in affects price and is closly related.

There are more relations that can be observed from this dataset, I did only small part of research in order to be able to make some conclusions from it.