Analysis of housing price in Ames, Iowa

Richard 徐龙昊 1630005056

Jemma 彭璨 1630005046

Sharon 袁亚雯 1630005062

Contents:

Abstract

For Ames Iowa housing price, this paper presents a reasonable model using multiple linear regression analysis. In the beginning, this paper states the background of the problem and the description of the data.

In preprocessing the data, for missing values, filled them according to the data type. At the same time, transformed the dependent variable to make its distribution closer to the normal distribution. Then, analyze the correlation coefficients of all numerical data and visuals them.

Then this paper proposes two main models to select variables; one uses the correlation coefficient, the other uses LASSO. For the first model, this paper directly deletes the variables with strong correlation coefficient and uses the remaining variables to do the model analysis. For the second model, this paper uses LASSO to select variables and then carries out further analysis. In this paper, multiple R-square, Adjusted R-square, t-statistic, F-statistic, PRESS and other statistical indicators are analyzed for the above two models, respectively. Using the independent test and non-constant variance test, check whether the assumptions of the multiple linear regression analysis are satisfied. The outliers, strong influence points and high leverage points are also analyzed. Then this paper improves the models using weighted least square and Box-Cox transformation and calculates their statistical indicators again.

Finally, by comparing the above models, this paper gives the suggested model in prediction analysis. In the end, this paper mentions some discussions and future works.

Keywords: multiple linear regression, LASSO, weighted least square, Box-Cox transformation

1. Description and Motivation

1.1 Background

It is universally acknowledged that housing price is one of the most heated discussed livelihood issues in recent years, which shows that people have grown increasingly concerned about the housing price. Specifically, this paper presents a data set describing the sale of almost every aspect of the individual residential property in Ames, Iowa from 2006 to 2010. This dataset is used to prove that much more influences price negotiations rather than simple common factors. Our data is related to 1460 observations, 79 variables, which focus on the quality and quantity of many physical attributes of the property, and the response is the housing price.

1.2 Dataset source

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
Id	MSSubClass	MSZoning	LotFrontage	LotArea	Street	Alley	LotShape	LandContou	Utilities	LotConfig	LandSlope	Neighborho	Condition1	Condition2	BldgType	HouseStyle	OverallQual	OverallCond	YearBuilt	YearRemodA
1	60	RL	65	8450	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	Norm	1Fam	2Story	7	5	2003	2003
2	20	RL	80	9600	Pave	NA	Reg	Lvl	AllPub	FR2	Gtl	Veenker	Feedr	Norm	1Fam	1Story	6	8	1976	1976
3	60	RL	68	11250	Pave	NA	IR1	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	Norm	1Fam	2Story	7	5	2001	2002
4	70	RL	60	9550	Pave	NA	IR1	Lvl	AllPub	Corner	Gtl	Crawfor	Norm	Norm	1Fam	2Story	7	5	1915	1970
5	60	RL	84	14260	Pave	NA	IR1	Lvl	AllPub	FR2	Gtl	NoRidge	Norm	Norm	1Fam	2Story	8	5	2000	2000
6	50	RL	85	14115	Pave	NA	IR1	Lvl	AllPub	Inside	Gtl	Mitchel	Norm	Norm	1Fam	1.5Fin	5	5	1993	1995
7	20	RL	75	10084	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	Somerst	Norm	Norm	1Fam	1Story	8	5	2004	2005
8	60	RL	NA	10382	Pave	NA	IR1	Lvl	AllPub	Corner	Gtl	NWAmes	PosN	Norm	1Fam	2Story	7	6	1973	1973
9	50	RM	51	6120	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	OldTown	Artery	Norm	1Fam	1.5Fin	7	5	1931	1950
10	190	RL	50	7420	Pave	NA	Reg	Lvl	AllPub	Corner	Gtl	BrkSide	Artery	Artery	2fmCon	1.5Unf	5	6	1939	1950
11	20	RL	70	11200	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	Sawyer	Norm	Norm	1Fam	1Story	5	5	1965	1965
12	60	RL	85	11924	Pave	NA	IR1	Lvl	AllPub	Inside	Gtl	NridgHt	Norm	Norm	1Fam	2Story	9	5	2005	2006
13	20	RL	NA	12968	Pave	NA	IR2	Lvl	AllPub	Inside	Gtl	Sawyer	Norm	Norm	1Fam	1Story	5	6	1962	1962
14	20	RL	91	10652	Pave	NA	IR1	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	Norm	1Fam	1Story	7	5	2006	2007
15	20	RL	NA	10920	Pave	NA	IR1	Lvl	AllPub	Corner	Gtl	NAmes	Norm	Norm	1Fam	1Story	6	5	1960	1960
16	45	RM	51	6120	Pave	NA	Reg	Lvl	AllPub	Corner	Gtl	BrkSide	Norm	Norm	1Fam	1.5Unf	7	8	1929	2001
17	20	RL	NA	11241	Pave	NA	IR1	Lvl	AllPub	CulDSac	Gtl	NAmes	Norm	Norm	1Fam	1Story	6	7	1970	1970
18	90	RL	72	10791	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	Sawyer	Norm	Norm	Duplex	1Story	4	5	1967	1967
19	20	RL	66	13695	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	SawyerW	RRAe	Norm	1Fam	1Story	5	5	2004	2004
20	20	RL	70	7560	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	NAmes	Norm	Norm	1Fam	1Story	5	6	1958	1965
21	60	RL	101	14215	Pave	NA	IR1	Lvl	AllPub	Corner	Gtl	NridgHt	Norm	Norm	1Fam	2Story	8	5	2005	2006
22	45	RM	57	7449	Pave	Grvl	Reg	Bnk	AllPub	Inside	Gtl	IDOTRR	Norm	Norm	1Fam	1.5Unf	7	7	1930	1950
23	20	RL	75	9742	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	CollgCr	Norm	Norm	1Fam	1Story	8	5	2002	2002
24	120	RM	44	4224	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	MeadowV	Norm	Norm	TwnhsE	1Story	5	7	1976	1976
25	20	RL	NA	8246	Pave	NA	IR1	Lvl	AllPub	Inside	Gtl	Sawyer	Norm	Norm	1Fam	1Story	5	8	1968	2001
26	20	RL	110	14230	Pave	NA	Reg	Lvl	AllPub	Corner	Gtl	NridgHt	Norm	Norm	1Fam	1Story	8	5	2007	2007
27	20	RL	60	7200	Pave	NA	Reg	Lvl	AllPub	Corner	Gtl	NAmes	Norm	Norm	1Fam	1Story	5	7	1951	2000
28	20	RL	98	11478	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	NridgHt	Norm	Norm	1Fam	1Story	8	5	2007	2008
29	20	RL	47	16321	Pave	NA	IR1	Lvl	AllPub	CulDSac	Gtl	NAmes	Norm	Norm	1Fam	1Story	5	6	1957	1997
30	30	RM	60	6324	Pave	NA	IR1	Lvl	AllPub	Inside	Gtl	BrkSide	Feedr	RRNn	1Fam	1Story	4	6	1927	1950
31	70	C (all)	50	8500	Pave	Pave	Reg	Lvl	AllPub	Inside	Gtl	IDOTRR	Feedr	Norm	1Fam	2Story	4	4	1920	1950
32	20	RL	NA	8544	Pave	NA	IR1	Lvl	AllPub	CulDSac	Gtl	Sawyer	Norm	Norm	1Fam	1Story	5	6	1966	2006
33	20	RL	85	11049	Pave	NA	Reg	Lvl	AllPub	Corner	Gtl	CollgCr	Norm	Norm	1Fam	1Story	8	5	2007	2007
34	20	RL	70	10552	Pave	NA	IR1	Lvl	AllPub	Inside	Gtl	NAmes	Norm	Norm	1Fam	1Story	5	5	1959	1959
35	120	RL	60	7313	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	NridgHt	Norm	Norm	TwnhsE	1Story	9	5	2005	2005
36	60	RL	108	13418	Pave	NA	Reg	Lvl	AllPub	Inside	Gtl	NridgHt	Norm	Norm	1Fam	2Story	8	5	2004	2005

```
> dim(data)
[1] 1460   81
> str(data)
'data.frame':  1460 obs. of  81 variables:
 $ Id           : int  1 2 3 4 5 6 7 8 9 10 ...
 $ MSSubClass   : int  60 20 60 70 60 50 20 60 50 190 ...
 $ MSZoning     : chr  "RL" "RL" "RL" "RL" ...
 $ LotFrontage  : int  65 80 68 60 84 85 75 NA 51 50 ...
 $ LotArea      : int  8450 9600 11250 9550 14260 14115 10084 10382 6120 7420 ...
 $ Street       : chr  "Pave" "Pave" "Pave" "Pave" ...
 $ Alley        : chr  NA NA NA NA ...
 $ LotShape     : chr  "Reg" "Reg" "IR1" "IR1" ...
 $ LandContour  : chr  "Lvl" "Lvl" "Lvl" "Lvl" ...
 $ Utilities    : chr  "AllPub" "AllPub" "AllPub" "AllPub" ...
 $ LotConfig    : chr  "Inside" "FR2" "Inside" "Corner" ...
 $ LandSlope    : chr  "Gtl" "Gtl" "Gtl" "Gtl" ...
 $ Neighborhood : chr  "CollgCr" "Veenker" "CollgCr" "Crawfor" ...
 $ Condition1   : chr  "Norm" "Feedr" "Norm" "Norm" ...
 $ Condition2   : chr  "Norm" "Norm" "Norm" "Norm" ...
 $ BldgType     : chr  "1Fam" "1Fam" "1Fam" "1Fam" ...
 $ HouseStyle   : chr  "2Story" "1Story" "2Story" "2Story" ...
 $ OverallQual  : int  7 6 7 7 8 5 8 7 7 5 ...
 $ OverallCond  : int  5 8 5 5 5 5 5 6 5 6 ...
 $ YearBuilt    : int  2003 1976 2001 1915 2000 1993 2004 1973 1931 1939 ...
 $ YearRemodAdd : int  2003 1976 2002 1970 2000 1995 2005 1973 1950 1950 ...
 $ RoofStyle    : chr  "Gable" "Gable" "Gable" "Gable" ...
 $ RoofMatl     : chr  "CompShg" "CompShg" "CompShg" "CompShg" ...
 $ Exterior1st  : chr  "VinylSd" "MetalSd" "VinylSd" "Wd Sdng" ...
 $ Exterior2nd  : chr  "VinylSd" "MetalSd" "VinylSd" "Wd Shng" ...
 $ MasVnrType   : chr  "BrkFace" "None" "BrkFace" "None" ...
 $ MasVnrArea   : int  196 0 162 0 350 0 186 240 0 0 ...
 $ ExterQual    : chr  "Gd" "TA" "Gd" "TA" ...
 $ ExterCond    : chr  "TA" "TA" "TA" "TA" ...
 $ Foundation   : chr  "PConc" "CBlock" "PConc" "BrkTil" ...
 $ BsmtQual     : chr  "Gd" "Gd" "Gd" "TA" ...
 $ BsmtCond     : chr  "TA" "TA" "TA" "Gd" ...
 $ BsmtExposure : chr  "No" "Gd" "Mn" "No" ...
 $ BsmtFinType1 : chr  "GLQ" "ALQ" "GLQ" "ALQ" ...
 $ BsmtFinSF1   : int  706 978 486 216 655 732 1369 859 0 851 ...
 $ BsmtFinType2 : chr  "Unf" "Unf" "Unf" "Unf" ...
 $ BsmtFinSF2   : int  0 0 0 0 0 0 0 32 0 0 ...
 $ BsmtUnfSF    : int  150 284 434 540 490 64 317 216 952 140 ...
 $ TotalBsmtSF  : int  856 1262 920 756 1145 796 1686 1107 952 991 ...
 $ Heating      : chr  "GasA" "GasA" "GasA" "GasA" ...
 $ HeatingQC    : chr  "Ex" "Ex" "Ex" "Gd" ...
 $ CentralAir   : chr  "Y" "Y" "Y" "Y" ...
```

```
> head(data)
  Id MSSubClass MSZoning LotFrontage LotArea Street Alley LotShape LandContour
1  1          60       RL          65    8450   Pave  <NA>      Reg         Lvl
2  2          20       RL          80    9600   Pave  <NA>      Reg         Lvl
3  3          60       RL          68   11250   Pave  <NA>      IR1         Lvl
4  4          70       RL          60    9550   Pave  <NA>      IR1         Lvl
5  5          60       RL          84   14260   Pave  <NA>      IR1         Lvl
6  6          50       RL          85   14115   Pave  <NA>      IR1         Lvl
  Utilities LotConfig LandSlope Neighborhood Condition1 Condition2 BldgType
1    AllPub    Inside       Gtl      CollgCr       Norm       Norm     1Fam
2    AllPub       FR2       Gtl      Veenker      Feedr       Norm     1Fam
3    AllPub    Inside       Gtl      CollgCr       Norm       Norm     1Fam
4    AllPub    Corner       Gtl      Crawfor       Norm       Norm     1Fam
5    AllPub       FR2       Gtl      NoRidge       Norm       Norm     1Fam
6    AllPub    Inside       Gtl      Mitchel       Norm       Norm     1Fam
  HouseStyle OverallQual OverallCond YearBuilt YearRemodAdd RoofStyle RoofMatl
1     2Story           7           5      2003         2003     Gable  CompShg
2     1Story           6           8      1976         1976     Gable  CompShg
3     2Story           7           5      2001         2002     Gable  CompShg
4     2Story           7           5      1915         1970     Gable  CompShg
5     2Story           8           5      2000         2000     Gable  CompShg
6     1.5Fin           5           5      1993         1995     Gable  CompShg
  Exterior1st Exterior2nd MasVnrType MasVnrArea ExterQual ExterCond Foundation
1     VinylSd     VinylSd    BrkFace        196        Gd        TA      PConc
2     MetalSd     MetalSd       None          0        TA        TA      CBlock
3     VinylSd     VinylSd    BrkFace        162        Gd        TA      PConc
4     Wd Sdng     Wd Shng       None          0        TA        TA      BrkTil
5     VinylSd     VinylSd    BrkFace        350        Gd        TA      PConc
6     VinylSd     VinylSd       None          0        TA        TA       Wood
  BsmtQual BsmtCond BsmtExposure BsmtFinType1 BsmtFinSF1 BsmtFinType2 BsmtFinSF2
1       Gd       TA           No          GLQ        706          Unf          0
2       Gd       TA           Gd          ALQ        978          Unf          0
3       Gd       TA           Mn          GLQ        486          Unf          0
4       TA       Gd           No          ALQ        216          Unf          0
5       Gd       TA           Av          GLQ        655          Unf          0
6       Gd       TA           No          GLQ        732          Unf          0
  BsmtUnfSF TotalBsmtSF Heating HeatingQC CentralAir Electrical X1stFlrSF
1       150         856    GasA        Ex          Y      SBrkr       856
2       284        1262    GasA        Ex          Y      SBrkr      1262
3       434         920    GasA        Ex          Y      SBrkr       920
4       540         756    GasA        Gd          Y      SBrkr       961
5       490        1145    GasA        Ex          Y      SBrkr      1145
6        64         796    GasA        Ex          Y      SBrkr       796
```

1.3 Dataset description

In dataset description, for variables, regarded the date data as categorical data since we assume that the house prices are relative stable.

The description of the variables in our data are as follows.

SalePrice - the property's sale price in dollars. This is the target variable that we aim to predict.

Categorical data

MSSubClass: The type of dwelling involved in the sale
MSZoning: The general zoning classification of the sale
Street: Type of road access to property
Alley: Type of alley access to property
LotShape: General shape of property
LandContour: Flatness of the property
Utilities: Type of utilities available
LotConfig: Lot configuration
LandSlope: Slope of property
Neighborhood: Physical locations within Ames city limits
Condition1: Proximity to main road or railroad
Condition2: Proximity to main road or railroad (if a second is present)
BldgType: Type of dwelling
HouseStyle: Style of dwelling
OverallQual: Overall material and finish quality
OverallCond: Overall condition rating
RoofStyle: Type of roof
RoofMatl: Roof material
Exterior1st: Exterior covering on house
Exterior2nd: Exterior covering on house (if more than one material)
MasVnrType: Masonry veneer type
ExterQual: Exterior material quality
ExterCond: Present condition of the material on the exterior
Foundation: Type of foundation
BsmtQual: Height of the basement
BsmtCond: General condition of the basement
BsmtExposure: Walkout or garden level basement walls
BsmtFinType1: Quality of basement finished area
BsmtFinType2: Quality of second finished area (if present)
Heating: Type of heating
HeatingQC: Heating quality and condition
CentralAir: Central air conditioning
Electrical: Electrical system
KitchenQual: Kitchen quality
Functional: Home functionality rating
FireplaceQu: Fireplace quality

GarageType: Garage location
GarageFinish: Interior finish of the garage
GarageQual: Garage quality
GarageCond: Garage condition
PavedDrive: Paved driveway
PoolQC: Pool quality
Fence: Fence quality
MiscFeature: Miscellaneous feature not covered in other categories
SaleType: Type of sale
SaleCondition: Condition of sale
MoSold: Month Sold
YrSold: Year Sold
GarageYrBlt: Year garage was built
YearBuilt: Original construction date
YearRemodAdd: Remodel date

Numerical data

LotFrontage: Linear feet of street connected to property
LotArea: Lot size in square feet
MasVnrArea: Masonry veneer area in square feet
BsmtFinSF1: Type 1 finished square feet
BsmtFinSF2: Type 2 finished square feet
BsmtUnfSF: Unfinished square feet of basement area
TotalBsmtSF: Total square feet of basement area
1stFlrSF: First Floor square feet
2ndFlrSF: Second floor square feet
LowQualFinSF: Low quality finished square feet (all floors)
GrLivArea: Above grade (ground) living area square feet
Bedroom: Number of bedrooms above basement level
Kitchen: Number of kitchens
TotRmsAbvGrd: Total rooms above grade (does not include bathrooms)
Fireplaces: Number of fireplaces
GarageCars: Size of garage in car capacity
GarageArea: Size of garage in square feet
WoodDeckSF: Wood deck area in square feet
OpenPorchSF: Open porch area in square feet
EnclosedPorch: Enclosed porch area in square feet
3SsnPorch: Three season porch area in square feet
ScreenPorch: Screen porch area in square feet
PoolArea: Pool area in square feet
MiscVal: $Value of miscellaneous feature
BsmtFullBath: Basement full bathrooms
BsmtHalfBath: Basement half bathrooms
FullBath: Full bathrooms above grade
HalfBath: Half baths above grade

2. Data Analysis

2.1 Data preprocessing

2.1.1 Missing data processing

Considering dataset from real-life will contain the missing data, we first visualize the missing data, and then visualize the ratio of the missing data. The results are following:

PoolQC	MiscFeature	Alley	Fence	FireplaceQu
0.9952054795	0.9630136986	0.9376712329	0.8075342466	0.4726027397
LotFrontage	GarageType	GarageYrBlt	GarageFinish	GarageQual
0.1773972603	0.0554794521	0.0554794521	0.0554794521	0.0554794521
GarageCond	BsmtExposure	BsmtFinType2	BsmtQual	BsmtCond
0.0554794521	0.0260273973	0.0260273973	0.0253424658	0.0253424658
BsmtFinType1	MasVnrType	MasVnrArea	Electrical	
0.0253424658	0.0054794521	0.0054794521	0.0006849315	

As the table shown, 19 variables contain the missing data, among them we have huge ratio of missing data up to 99%+ and small ratio low to 0.0685%, which obviously exists significant differences. Now we back to continue further analysis.

In order to ensure the integrity of data, we now impute all the missing data based on different situations and features. For the majority of categorical data, NA can be replaced by none; for some of categorical data, NA can be replaced by majority of other values in this set; for numerical data, missing data can be replaced by median under features. After all procedures are done, we should double check if there is any missing data still exists.

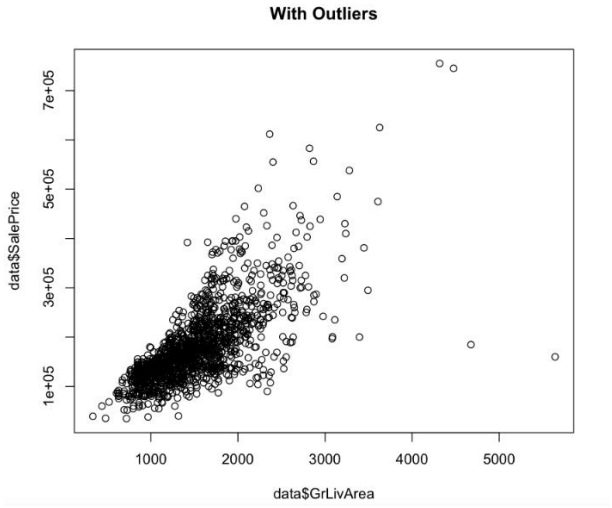
Imputing missing data:

1. PoolQC: NA in data means "No Pool". Since the ratio of missing value to (99%+) is huge and majority of houses have no Pool in common situation.
2. MiscFeature: NA in data means "no misc feature".
3. Alley: NA in data means "no alley access".
4. Fence: NA in data means "no fence".
5. FireplaceQu: NA in data means "no fireplace".
6. LotFrontage: Since the area of each street connected to the house property most likely have a similar area to its' neighborhood; missing values can be replaced by the median LotFrontage of the neighborhood.
7. GarageType: let 'None' replace the missing data.
8. GarageYrBlt:l let 0 replace the missing data.
9. GarageFinish: let 'None' replace the missing data.
10. GarageQual: let 'None' replace the missing data.
11. GarageCond: Use 'None' to replace missing data in this variable.
12. BsmtExposure: For the categorical feature BsmtExposure, 'NA' is used to denote that there is no basement.
13. BsmtFinType2: For the categorical feature BsmtFinType2, 'NA' is used to denote that there is no basement.
14. BsmtQual: For the categorical feature BsmtQual, 'NA' is used to denote that

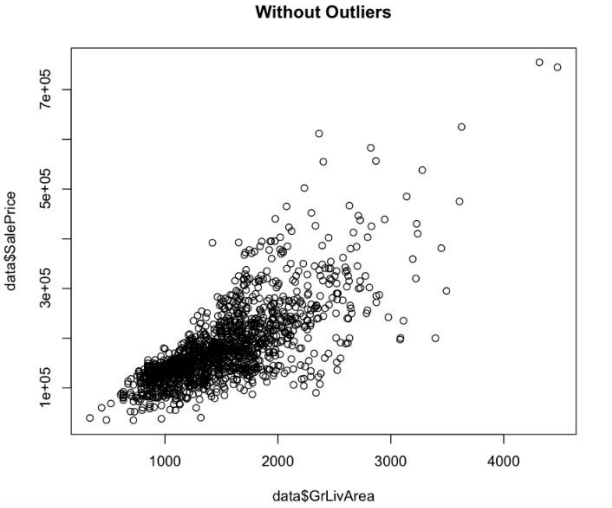
there is no basement.

15. BsmtCond: For the categorical feature BsmtCond, ‘NA’ is used to denote that there is no basement.
16. BsmtFinType1: For the categorical feature BsmtFinType1, ‘NA’ is used to denote that there is no basement.
17. MasVnrType: ‘NA’ in this variable mostly likely means that there is no masonry veneer for these houses, so we fill ‘None’ in it.
18. MasVnrArea: ‘NA’ in this variable mostly likely means that there is no masonry veneer for these houses, so we fill 0 in it.
19. Electrical: There is only one ‘NA’ value of the variable Electrical, and since this feature has the highest ‘SBrkr’, we can set ‘SBrkr’ in this missing value.

2.1.2 Non-informative outlier detection



As is shown in the above plot, we can see that there are two unreasonable cases of the bottom right of the plot. Since it is impractical that an extreme large above ground living area with a low price, we directly delete two outliers to enhance the model.

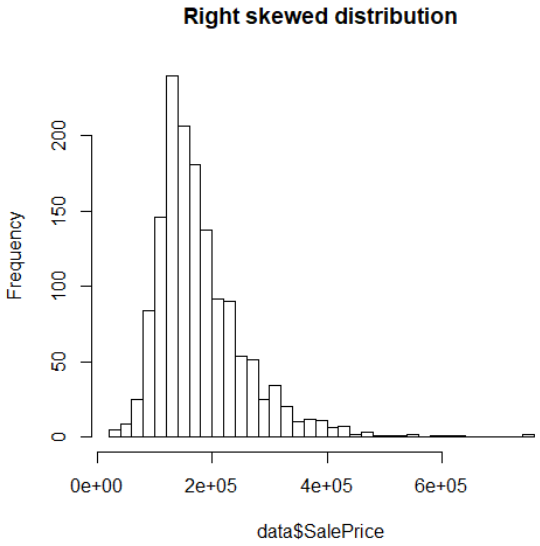


The reason why we only manage to delete the outliers in the above case is that the

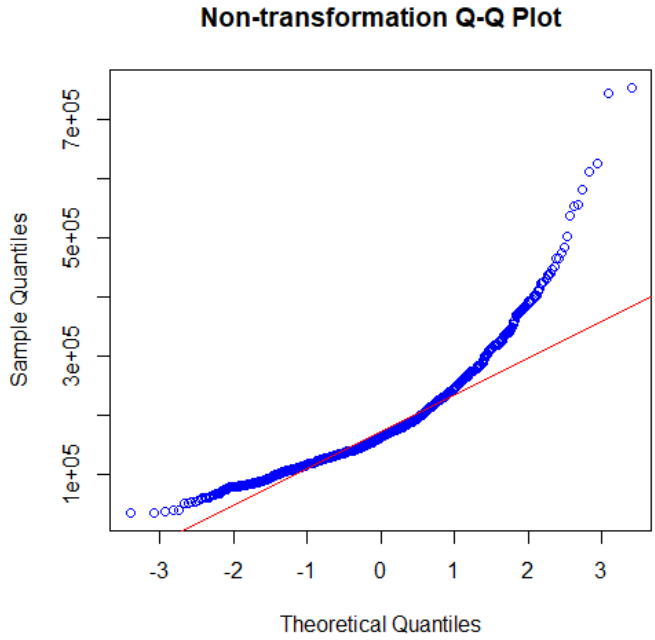
outliers of above ground living area will strongly affect the significance of model. However, it is unnecessary to delete all the outliers in the variable because we need to consider the practical data in order to enhance the ability of predicting unknown data. In this way, we are capable of promoting model.

2.1.3 Response variable analysis

SalePrice is the variable we want to predict. We first analyze the collected data onto SalePrice and if necessary, try to transform it and get the normally distributed data which can be useful in multiple linear model.

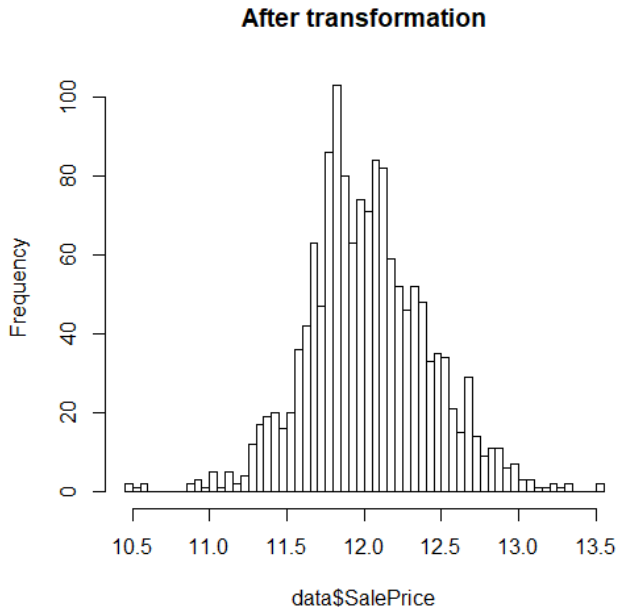


We begin with the SalePrice distribution, as the plot shows a strong right-skewed distribution rather than normal distribution, we next test the normality of the SalePrice.

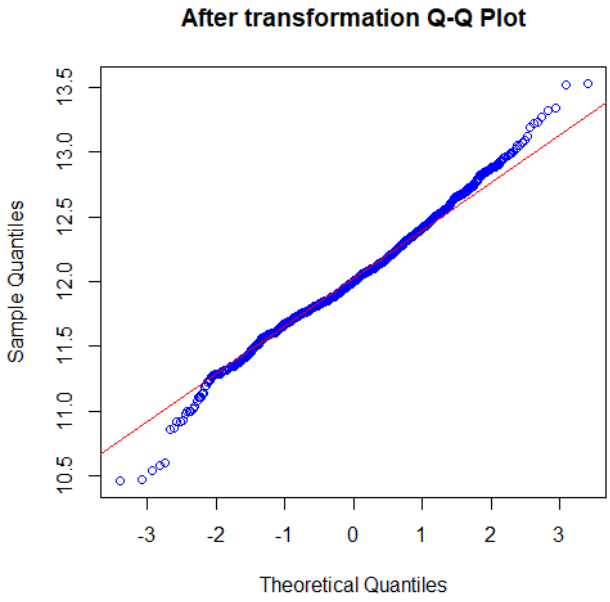


The Q-Q plot to enhance the proof that the SalePrice distribution is not normal distribution. We need to transform the data onto normally distributed data which can be contributed to linear models later.

After transformation by log(y+1), we analyze the new data onto y.



After transformation, we can observe from the plot that SalePrice distribution is more likely to be the normal contribution than before. We here regard the unimodal symmetry as approximate normal distribution. We still need to test the normality of the SalePrice after transformation.



The Q-Q plot to enhance the proof that the SalePrice distribution after transformation is closer to being the normal distribution.

2.1.5 Correlation analysis

Among all numerical variables, let every two variables of them be one pair, we want to compute the correlation between each pair. If we use cor() directly, we will get the following results.

	LotFrontage	LotArea	MasVnrArea	BsmtFinSF1
LotFrontage	1.0000000000	0.30452217	0.17847562	0.214366614
LotArea	0.3045221707	1.00000000	0.10332756	0.214103131
MasVnrArea	0.1784756230	0.10332756	1.00000000	0.261280710
BsmtFinSF1	0.2143666136	0.21410313	0.26128071	1.000000000
BsmtFinSF2	0.0424632280	0.11116975	-0.07133961	-0.050117400
BsmtUnfSF	0.1240982099	-0.00261836	0.11386624	-0.495251469
TotalBsmtSF	0.3634723116	0.26083313	0.36009343	0.522396052
X1stFlrSF	0.4137725324	0.29947458	0.33987558	0.445862656
X2ndFlrSF	0.0723880208	0.05098595	0.17380360	-0.137078986
LowQualFinSF	0.0374693108	0.00477897	-0.06863281	-0.064502597
GrLivArea	0.3680074392	0.26311617	0.38807312	0.208171130
BsmtFullBath	0.0903428654	0.15815453	0.08303046	0.649211754
BsmtHalfBath	-0.0069789943	0.04804557	0.02739582	0.067418478
FullBath	0.1805337932	0.12603063	0.27302751	0.058543137
HalfBath	0.0472216405	0.01425947	0.19912556	0.004262424
BedroomAbvGr	0.2368400350	0.11968991	0.10277196	-0.107354677
KitchenAbvGr	-0.0049046705	-0.01778387	-0.03843998	-0.081006851
Fireplaces	0.2332206400	0.27136401	0.24702598	0.260010920
GarageCars	0.2695392541	0.15487074	0.36196533	0.224053522
GarageArea	0.3235109288	0.18040276	0.37090355	0.296970385
WoodDeckSF	0.0755421091	0.17169769	0.15998572	0.204306145
OpenPorchSF	0.1370135606	0.08477381	0.12255286	0.111760613

The closer of the result to number +1 or -1, the stronger correlation represents. As this method is not presenting different level of correlation directly. We will use another way to give a more visual representation of the same result. We choose to draw the matrix thermodynamic diagram.

2.2 Methodology

1. t-statistics:
   Null hypothesis: The variable is significant.
   Alternative hypothesis: The variable is not significant.
   The level of significance: 0.01
2. F-statistic:
   Null hypothesis: All beta equals to zero.
   Alternative hypothesis: At least some beta does not equal to zero.
   The level of significance: 0.01
3. Durbin-Watson Test:
   Null hypothesis: There is no first-order serial correlation for random error terms.
   Alternative hypothesis: There is first-order serial correlation for random error terms.
   The level of significance: 0.05
4. NCV Test:
   Null hypothesis: The error variance remains unchanged.
   Alternative hypothesis: Error variance varies with the level of fitting value.

The level of significance: 0.05

5. Kolmogorov-Smirnov Test:
   Null hypothesis: The data follows the theoretical distribution.
   Alternative hypothesis: The data not follows the practical distribution.
   The level of significance: 0.05
6. Variance inflation factor:
   When 0 < VIF < 5, there is no multicollinearity.
   When 5 < VIF < 10, there is weak multicollinearity.
   When 10 < VIF < 100, there is strong multicollinearity.
   When VIF > 100, there is severe multicollinearity

2.3 Modeling

2.3.1. Drop using correlation

Initially, we attempt to build Multiple linear regression model only consider 28 numerical variables.

```
Call:
lm(formula = y ~ ., data = data_frame_lm)

Residuals:
     Min       1Q   Median       3Q      Max
-0.97759 -0.07566  0.01752  0.09776  0.61135

Coefficients: (2 not defined because of singularities)
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.112e+01  3.016e-02 368.620  < 2e-16 ***
LotFrontage    9.912e-04  2.404e-04   4.123 3.96e-05 ***
LotArea        7.039e-07  4.887e-07   1.440 0.149965
MasVnrArea     9.021e-05  5.006e-05   1.802 0.071775 .
BsmtFinSF1     2.830e-04  2.272e-05  12.455  < 2e-16 ***
BsmtFinSF2     1.808e-04  3.408e-05   5.305 1.30e-07 ***
BsmtUnfSF      2.256e-04  1.951e-05  11.566  < 2e-16 ***
TotalBsmtSF           NA         NA      NA       NA
X1stFlrSF      2.346e-04  2.764e-05   8.488  < 2e-16 ***
X2ndFlrSF      2.405e-04  2.238e-05  10.747  < 2e-16 ***
LowQualFinSF  -8.032e-05  9.362e-05  -0.858 0.391044
GrLivArea             NA         NA      NA       NA
BsmtFullBath   5.338e-02  1.252e-02   4.265 2.13e-05 ***
BsmtHalfBath   2.841e-02  1.966e-02   1.446 0.148536
FullBath       1.336e-01  1.172e-02  11.400  < 2e-16 ***
HalfBath       7.233e-02  1.205e-02   6.002 2.46e-09 ***
BedroomAbvGr  -4.782e-02  7.919e-03  -6.038 1.98e-09 ***
KitchenAbvGr  -2.614e-01  2.243e-02 -11.650  < 2e-16 ***
TotRmsAbvGrd   1.765e-02  5.927e-03   2.979 0.002944 **
Fireplaces     3.704e-02  8.465e-03   4.375 1.30e-05 ***
GarageCars     8.494e-02  1.372e-02   6.192 7.74e-10 ***
GarageArea     1.110e-04  4.719e-05   2.352 0.018786 *
WoodDeckSF     1.368e-04  3.853e-05   3.551 0.000396 ***
OpenPorchSF    1.390e-04  7.308e-05   1.902 0.057342 .
EnclosedPorch -2.247e-04  7.665e-05  -2.932 0.003425 **
X3SsnPorch     1.578e-04  1.517e-04   1.040 0.298553
ScreenPorch    1.694e-04  8.267e-05   2.049 0.040671 *
PoolArea       9.824e-06  1.179e-04   0.083 0.933576
MiscVal        4.294e-06  8.948e-06   0.480 0.631394
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1681 on 1431 degrees of freedom
Multiple R-squared:  0.8262,   Adjusted R-squared:  0.8231
F-statistic: 261.7 on 26 and 1431 DF,  p-value: < 2.2e-16
```

As it shows in the result, there are missing values 'NA' locate in variable "TotalBsmtSF" and "GrLivArea", which means the matrix is singular and gives no solution.

Then we use R command kappa() to get the conditional number of matrix:

```
[1] 2.026486e+19
```

Since the result is greater than 1000, it shows a severe multicollinearity. Therefore, we cannot do subsequent operations based on this model.

Then we raise the idea of dropping highly correlated variables. As it mentioned in the correlation matrix, if there exist two variables which are highly correlated, we drop the one with lower correlation with the SalePrice. In this way, we delete the variable BsmtFinSF1, TotalBsmtSF, GrLivArea, TotRmsAbvGrd, and GarageArea. Then we build the Model 1.

y ~ LotFrontage + LotArea + MasVnrArea + BsmtFinSF2 + BsmtUnfSF +
X1stFlrSF + X2ndFlrSF + LowQualFinSF + BsmtFullBath + BsmtHalfBath +
FullBath + HalfBath + BedroomAbvGr + KitchenAbvGr + Fireplaces +
GarageCars + WoodDeckSF + OpenPorchSF + EnclosedPorch + X3SsnPorch +
ScreenPorch + PoolArea + MiscVal

We find the following variables have a strong influence on the response variable based on this model.

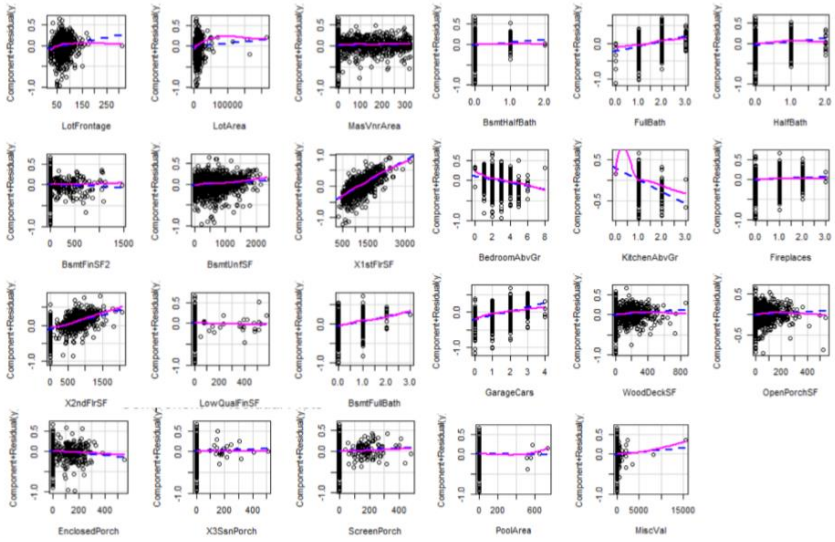
HalfBath: Half baths above grade
LotArea: Lot size in square feet
BsmtHalfBath: Basement half bathrooms
BsmtUnfSF: Unfinished square feet of basement area
1stFlrSF: First Floor square feet
BsmtFinSF2: Type 2 finished square feet
LowQualFinSF: Low quality finished square feet (all floors)

The summary of Model 1 is as follows:

```
Call:
lm(formula = y ~ ., data = as.data.frame(data_drop))

Residuals:
     Min       1Q   Median       3Q      Max
-0.94191 -0.08281  0.01763  0.10727  0.66317

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.116e+01  3.152e-02 354.238  < 2e-16 ***
LotFrontage    1.099e-03  2.527e-04   4.348 1.47e-05 ***
LotArea        7.479e-07  5.159e-07   1.450 0.147350
MasVnrArea     1.322e-04  5.273e-05   2.508 0.012261 *
BsmtFinSF2    -4.410e-05  3.061e-05  -1.441 0.149941
BsmtUnfSF      5.026e-05  1.425e-05   3.527 0.000433 ***
X1stFlrSF      4.464e-04  2.166e-05  20.609  < 2e-16 ***
X2ndFlrSF      2.634e-04  1.992e-05  13.221  < 2e-16 ***
LowQualFinSF  -4.965e-05  9.754e-05  -0.509 0.610828
BsmtFullBath   1.250e-01  1.177e-02  10.626  < 2e-16 ***
BsmtHalfBath   6.427e-02  2.050e-02   3.136 0.001750 **
FullBath       1.400e-01  1.231e-02  11.374  < 2e-16 ***
HalfBath       8.199e-02  1.266e-02   6.476 1.29e-10 ***
BedroomAbvGr  -4.218e-02  7.455e-03  -5.657 1.85e-08 ***
KitchenAbvGr  -2.949e-01  2.258e-02 -13.063  < 2e-16 ***
Fireplaces     3.428e-02  8.871e-03   3.864 0.000116 ***
GarageCars     1.201e-01  8.271e-03  14.517  < 2e-16 ***
WoodDeckSF     1.560e-04  4.066e-05   3.836 0.000130 ***
OpenPorchSF    1.946e-04  7.686e-05   2.532 0.011445 *
EnclosedPorch -2.558e-04  8.085e-05  -3.164 0.001591 **
X3SsnPorch     1.113e-04  1.602e-04   0.695 0.487093
ScreenPorch    2.067e-04  8.719e-05   2.371 0.017891 *
PoolArea      -3.162e-05  1.241e-04  -0.255 0.798980
MiscVal        1.074e-05  9.429e-06   1.139 0.254762
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1776 on 1434 degrees of freedom
Multiple R-squared:  0.8058,   Adjusted R-squared:  0.8026
F-statistic: 258.6 on 23 and 1434 DF,  p-value: < 2.2e-16
```

To interpret this result, the right-hand side ‘*’ sign is used to visualize the p-value of the t test for each variable. The Multiple R-squared shows that 80.58% variability of Housing price can be explained by this model. After balancing the fitness and model complexity, we get Adjusted R-square 0.8026. For the F-statistic, since the p-value is less than 0.01, we reject the null hypothesis, which gives the conclusion that the model is significant.

To learn the relationship between y(SalePrice) and 23 variables, we test the non-linearity in the all 23 regressors. The component-plus-resident plots are following:



The local linear fitting using span = 0.5 and the linear least squares line are plotted in each picture. The span range is small, so we can conclude that all 23 variables are linear.

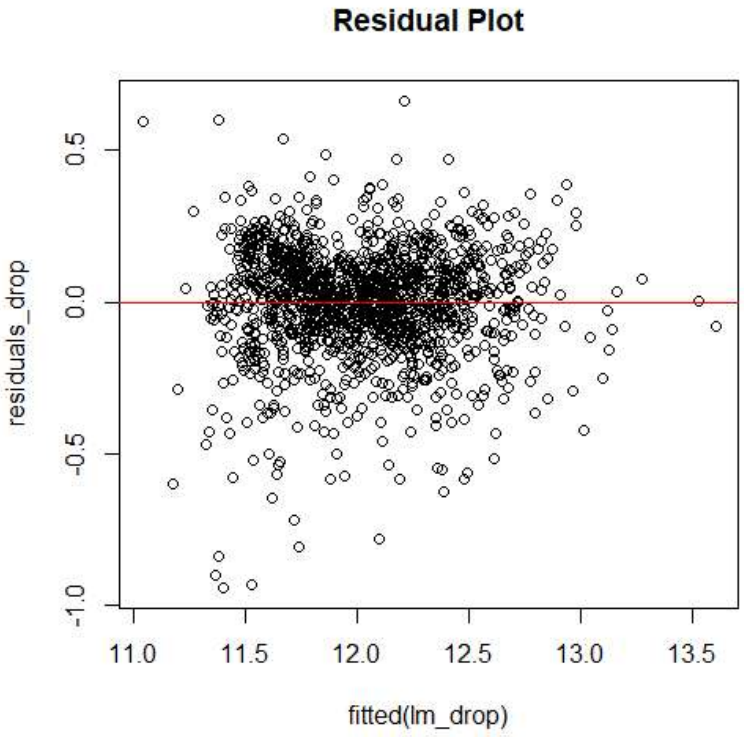
The criteria donated by the PRESS statistics in this variable selection will be as follows: `[1] 47.37008`

Then we calculate the Variance Inflation Factor, the result are as follows:

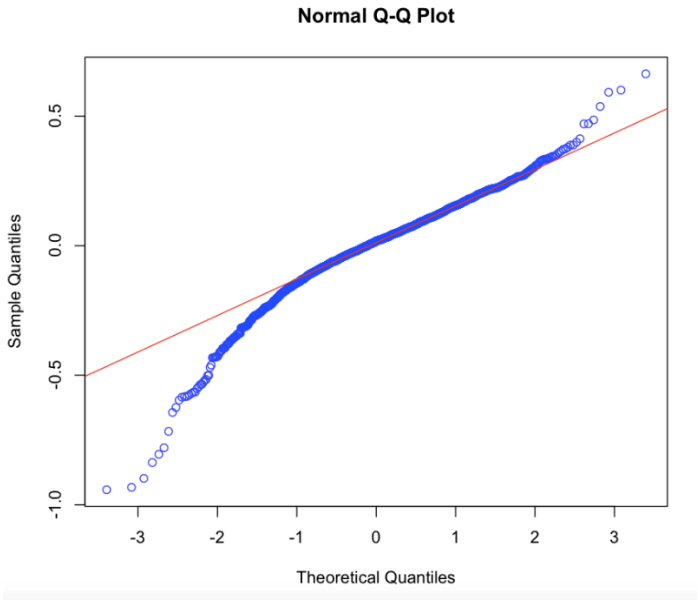
LotFrontage	LotArea	MasVnrArea	BsmtFinSF2	BsmtUnfSF	X1stFlrSF
1.306768	1.195485	1.205113	1.128497	1.833210	3.001333
X2ndFlrSF	LowQualFinSF	BsmtFullBath	BsmtHalfBath	FullBath	HalfBath
3.477876	1.040807	1.712910	1.107900	2.117338	1.871731
BedroomAbvGr	KitchenAbvGr	Fireplaces	GarageCars	WoodDeckSF	OpenPorchSF
1.711431	1.145152	1.498726	1.764423	1.200309	1.164505
EnclosedPorch	X3SsnPorch	ScreenPorch	PoolArea	MiscVal	
1.129794	1.020116	1.093594	1.039508	1.012499	

As we can see, all the values are between 0 and 10, which indicates that there is no severe collinearity in this model.

Additionally, we perform the residual plot and Q-Q Plot



The residual plot tends to be a Left-opening megaphone, which suggests that variance decreasing with the quality plotted on the x-axis.



As we can see from the Q-Q plot, the tail from both side tends to deviate the reference line, it seems that the curve does not suppose normality.

We here use durbinWatsonTest to achieve independent test:

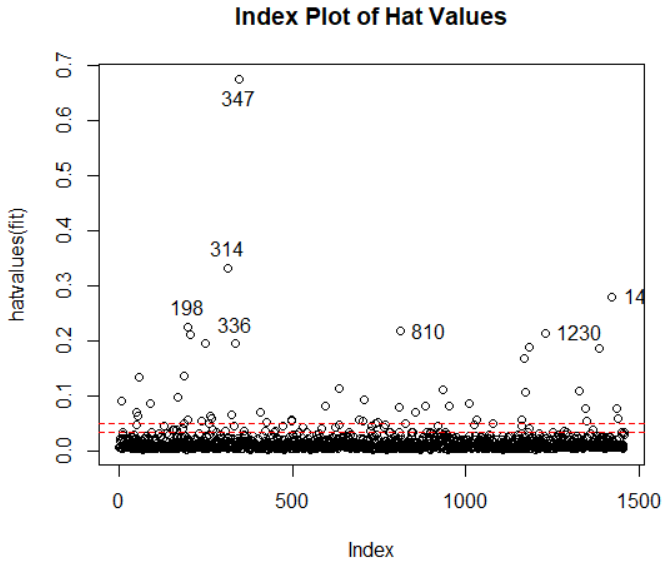
```
lag Autocorrelation D-W Statistic p-value
  1    -0.005488564      2.010746     0.9
Alternative hypothesis: rho != 0
```

For the durbinWatson Test, since the P-value is larger than the level of significance, we cannot reject the null hypothesis. Therefore, the residuals are independent.

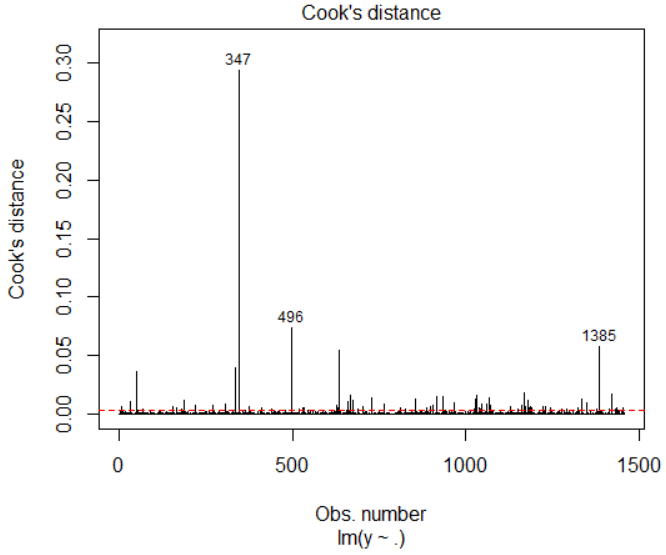
```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 26.32051, Df = 1, p = 2.892e-07
```

Then we use the NCV test to measure the heteroscedasticity. Since the P-value is less than 0.05, we reject the null hypothesis. Therefore, there exists a heteroscedasticity, the model is unstable.

Using the leverage value to detect the model and finding the outliers. The high leverage value is decided by the hat statistics where the hat value is greater two or three times than the mean of hat value.



Vertical reference lines are drawn at twice and three times the average hat value. Any observation above this two line has a potential for exerting strong influence on the result.



We can see that there are several observations which have high Cook's distance

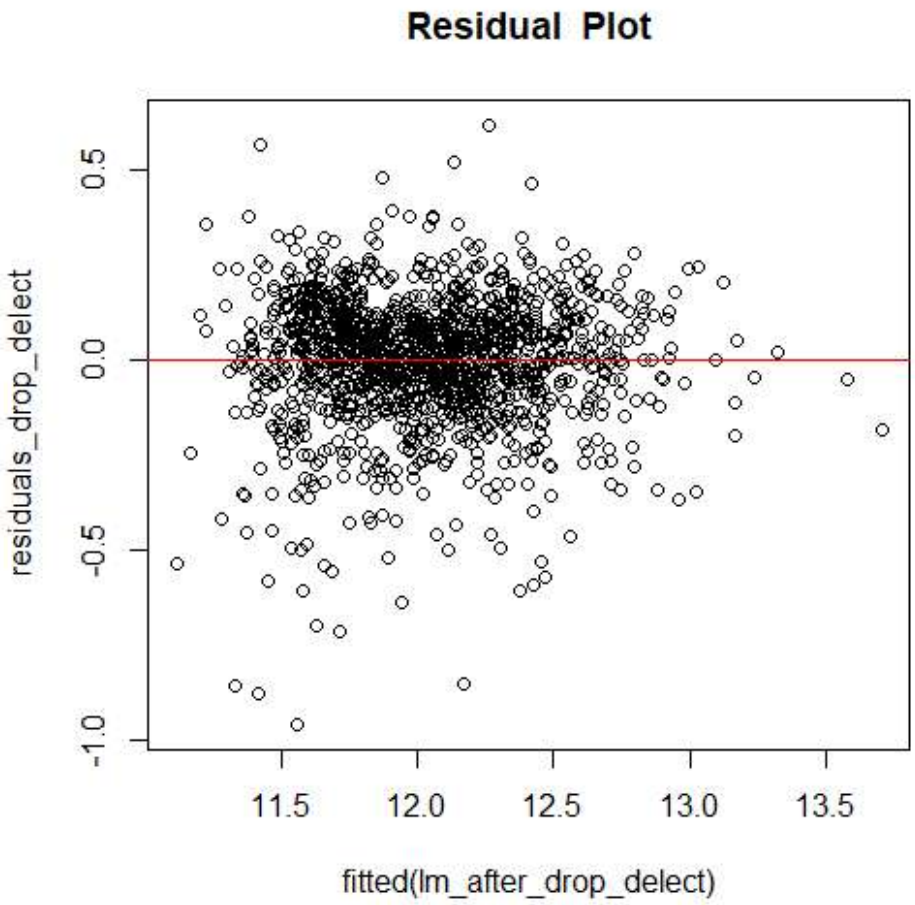
values. In some situations, these observations that with high influential feature may potentially exist abnormal influences just like outlier's feature. The Cook's distance plot only identifies the influential points rather than giving reasons. We will compare and find out the outliers and influential points with comprehensive consideration. Therefore, we further consider the intersection of the above two measures.
Then we found six observations:

```
[1]   49  496  635  854 1031 1385
```

We delete the above six observations and get the following model:

```
Call:
lm(formula = y[-drop_delect, ] ~ ., data = as.data.frame(data_after_drop_delect))

Residuals:
     Min       1Q   Median       3Q      Max
-0.93444 -0.08478  0.01782  0.10650  0.66815

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.119e+01  3.121e-02 358.342  < 2e-16 ***
LotFrontage    1.159e-03  2.477e-04   4.679 3.15e-06 ***
LotArea        7.476e-07  5.046e-07   1.481 0.138731
MasVnrArea     1.249e-04  5.161e-05   2.421 0.015599 *
BsmtFinSF2    -3.538e-05  3.035e-05  -1.166 0.243860
BsmtUnfSF      4.359e-05  1.396e-05   3.121 0.001836 **
X1stFlrSF      4.432e-04  2.124e-05  20.866  < 2e-16 ***
X2ndFlrSF      2.526e-04  1.954e-05  12.927  < 2e-16 ***
LowQualFinSF  -1.009e-04  1.013e-04  -0.996 0.319631
BsmtFullBath   1.218e-01  1.153e-02  10.565  < 2e-16 ***
BsmtHalfBath   5.996e-02  2.006e-02   2.990 0.002840 **
FullBath       1.417e-01  1.210e-02  11.711  < 2e-16 ***
HalfBath       8.505e-02  1.240e-02   6.857 1.05e-11 ***
BedroomAbvGr  -4.114e-02  7.379e-03  -5.575 2.95e-08 ***
KitchenAbvGr  -3.187e-01  2.282e-02 -13.969  < 2e-16 ***
Fireplaces     3.422e-02  8.688e-03   3.938 8.60e-05 ***
GarageCars     1.202e-01  8.115e-03  14.811  < 2e-16 ***
WoodDeckSF     1.520e-04  3.982e-05   3.816 0.000141 ***
OpenPorchSF    3.088e-04  7.811e-05   3.953 8.10e-05 ***
EnclosedPorch -2.321e-04  7.934e-05  -2.925 0.003498 **
X3SsnPorch     1.119e-04  1.567e-04   0.714 0.475142
ScreenPorch    2.609e-04  8.742e-05   2.984 0.002893 **
PoolArea       7.108e-05  1.297e-04   0.548 0.583749
MiscVal        1.341e-05  9.266e-06   1.447 0.148148
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1737 on 1428 degrees of freedom
Multiple R-squared:  0.8127,   Adjusted R-squared:  0.8097
F-statistic: 269.4 on 23 and 1428 DF,  p-value: < 2.2e-16
```

The Multiple R-squared means that 81.27% variability of Housing price can be explained by this model. After balancing the fitness and model complexity, we get Adjusted R-square 0.8097. For the F-statistic, since the p-value is less than 0.01, we reject the null hypothesis, which gives the conclusion that the model is significant.



As it shows in the above residual plot, it still tends to be a Left-opening megaphone, which suggests that variance decreasing with the quality plotted on the x-axis. So we further consider the ncvTest to identify the homoscedasticity.

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 27.48219, Df = 1, p = 1.5855e-07
```

Since the P-value is still less than 0.05, we reject the null hypothesis. Therefore, there exists a heteroscedasticity, the model seems unstable.

Due to the non-constant variance violates the assumptions of linear regression model. When non-constant variance is diagnosed, while exact variances are unknown, we could contemplate two remedies: Weighted least square and Box-cox transformation.

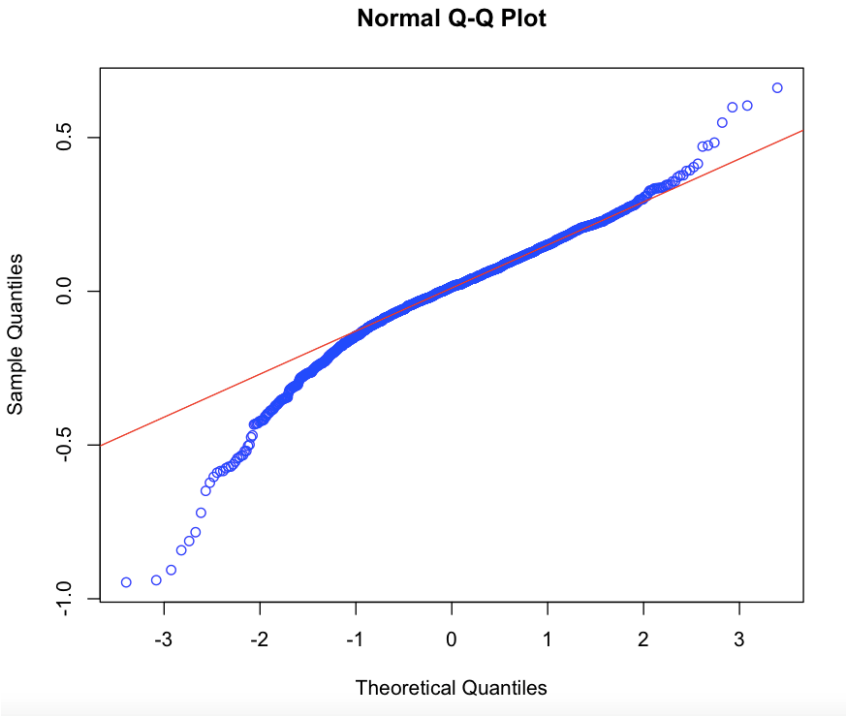
To start with, we apply the weighted least squares. Here we use the reciprocal of the absolute residual value as a weight to modify the model.

```
Call:
lm(formula = y ~ ., data = as.data.frame(data_drop), weights = 1/abs(residuals_drop))

Weighted Residuals:
    Min      1Q  Median      3Q     Max
-0.9754 -0.2898  0.1094  0.3194  0.8125

Coefficients:
               Estimate Std. Error  t value Pr(>|t|)
(Intercept)    1.118e+01  1.109e-02 1008.604  < 2e-16 ***
LotFrontage    1.049e-03  1.117e-04    9.392  < 2e-16 ***
LotArea        1.023e-06  2.500e-07    4.091 4.53e-05 ***
MasVnrArea     1.449e-04  1.769e-05    8.193 5.59e-16 ***
BsmtFinSF2    -4.524e-05  1.217e-05   -3.716  0.00021 ***
BsmtUnfSF      4.866e-05  5.184e-06    9.386  < 2e-16 ***
X1stFlrSF      4.432e-04  7.473e-06   59.314  < 2e-16 ***
X2ndFlrSF      2.630e-04  6.635e-06   39.637  < 2e-16 ***
LowQualFinSF  -6.021e-05  4.674e-05   -1.288  0.19785
BsmtFullBath   1.236e-01  4.221e-03   29.276  < 2e-16 ***
BsmtHalfBath   6.483e-02  6.868e-03    9.439  < 2e-16 ***
FullBath       1.392e-01  4.689e-03   29.678  < 2e-16 ***
HalfBath       7.721e-02  4.547e-03   16.980  < 2e-16 ***
BedroomAbvGr  -4.246e-02  3.135e-03  -13.546  < 2e-16 ***
KitchenAbvGr  -3.003e-01  7.335e-03  -40.938  < 2e-16 ***
Fireplaces     3.135e-02  2.981e-03   10.515  < 2e-16 ***
GarageCars     1.216e-01  3.046e-03   39.902  < 2e-16 ***
WoodDeckSF     1.600e-04  1.432e-05   11.175  < 2e-16 ***
OpenPorchSF    1.923e-04  2.759e-05    6.969 4.85e-12 ***
EnclosedPorch -2.546e-04  2.470e-05  -10.309  < 2e-16 ***
X3SsnPorch     7.327e-05  5.298e-05    1.383  0.16685
ScreenPorch    2.021e-04  3.962e-05    5.101 3.84e-07 ***
PoolArea      -2.108e-05  5.986e-05   -0.352  0.72480
MiscVal        9.168e-06  6.745e-06    1.359  0.17431
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3604 on 1434 degrees of freedom
Multiple R-squared:  0.9851, Adjusted R-squared:  0.9848
F-statistic:  4109 on 23 and 1434 DF,  p-value: < 2.2e-16
```

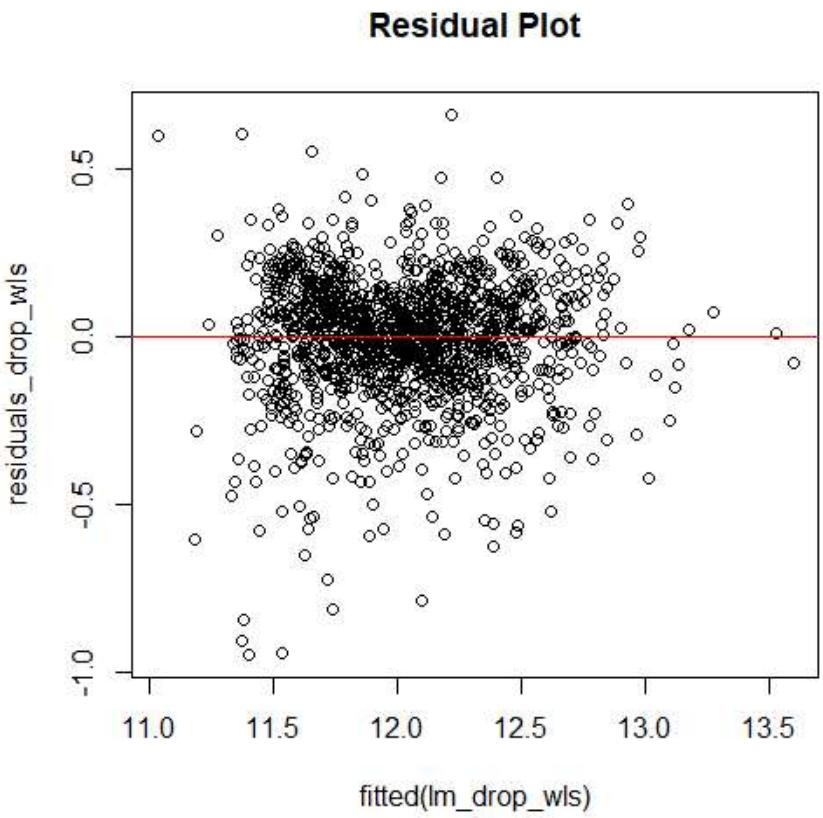
As is shown in above result, we can visually find that the model has been improved by ‘***’ sign. Both multiple R-squared and Adjusted R-squared has raised to 0.9851 and 0.9848, it shows that 98.51% variability of Housing price can be explained by this model now. For the F-statistic, the P-value is less than level of significance 0.01, we can have the statistical evidence that the model is significant.

The criteria donated by the PRESS statistics in this variable selection will be as follows: `[1] 45.60279`

Additionally, we test the normality and homoscedasticity by normal residuals plot, normal Q-Q plot and the Kolmogorov-Smirnov test.



As we can see from the Q-Q Plot, it seems that the curve does not suppose normality.



As it shows in the above residues plot, it tends to improve a slight level but still similar as left-opening megaphone.

We use ncvTest to identify the homoscedasticity:

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 2.422245, Df = 1, p = 0.11962
```

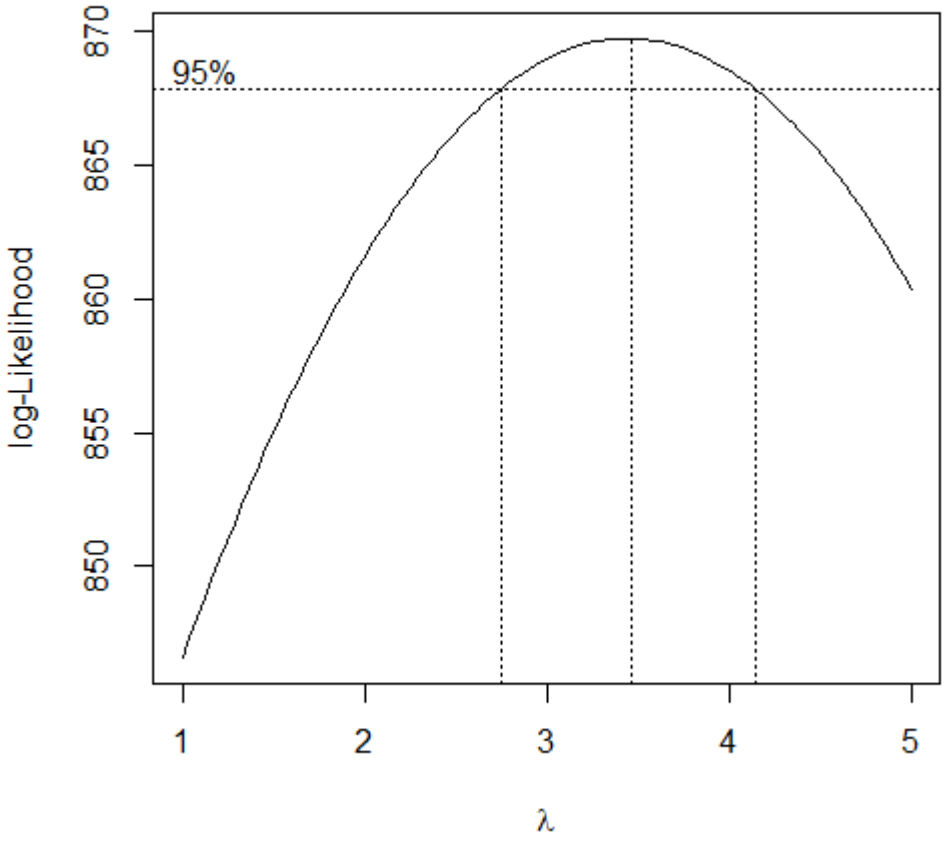
As is shown in the NCV test, since the P-value is 0.11962, greater than 0.05, we cannot reject the null hypothesis. Therefore, we have the statistical evidence that variance is a constant.

Here we use the Kolmogorov-Smirnov test to test whether the data follow the normal distribution.

```
    One-sample Kolmogorov-Smirnov test

data:  jitter(residuals_drop_wls)
D = 0.074898, p-value = 1.574e-07
alternative hypothesis: two-sided
```

As is shown in the result, the p-value is smaller than 0.05 that prove the non-normality.

Another way, we further consider the Box-cox transformation. It starts from the determination of the λ based on the log maximum likelihood function. The plot shows the procedure and the final λ are follows:



```
> powerTransform(lm_drop)$lambda
      Y1
3.448519
```

Then we get the value of λ, and substitute into Box-cox transformation.

$$y(\lambda) = \begin{cases} \dfrac{y^{3.448591} - 1}{3.448591}, \lambda \neq 0 \\ \ln y, \quad \lambda = 0 \end{cases}$$

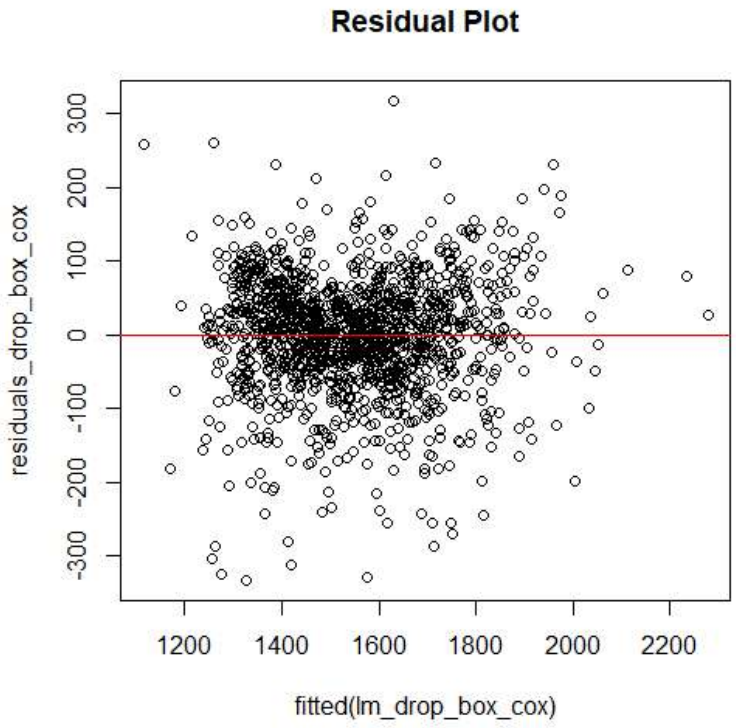
The brief results after transformation as follows:

```
Call:
lm(formula = y_drop_box_cox ~ ., data = as.data.frame(data_drop))

Residuals:
    Min      1Q  Median      3Q     Max
-333.46  -37.54    5.60   44.81  318.59

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.167e+03  1.366e+01  85.423  < 2e-16 ***
LotFrontage    5.107e-01  1.096e-01   4.662 3.43e-06 ***
LotArea        3.989e-04  2.237e-04   1.783 0.074726 .
MasVnrArea     6.670e-02  2.286e-02   2.917 0.003585 **
BsmtFinSF2    -2.504e-02  1.327e-02  -1.886 0.059470 .
BsmtUnfSF      2.109e-02  6.177e-03   3.414 0.000658 ***
X1stFlrSF      2.121e-01  9.392e-03  22.580  < 2e-16 ***
X2ndFlrSF      1.302e-01  8.638e-03  15.074  < 2e-16 ***
LowQualFinSF  -1.921e-02  4.229e-02  -0.454 0.649736
BsmtFullBath   5.414e+01  5.102e+00  10.612  < 2e-16 ***
BsmtHalfBath   2.593e+01  8.886e+00   2.919 0.003570 **
FullBath       5.884e+01  5.337e+00  11.026  < 2e-16 ***
HalfBath       3.298e+01  5.489e+00   6.009 2.36e-09 ***
BedroomAbvGr  -2.338e+01  3.232e+00  -7.233 7.64e-13 ***
KitchenAbvGr  -1.318e+02  9.789e+00 -13.466  < 2e-16 ***
Fireplaces     1.305e+01  3.846e+00   3.393 0.000709 ***
GarageCars     5.109e+01  3.586e+00  14.247  < 2e-16 ***
WoodDeckSF     6.867e-02  1.763e-02   3.896 0.000102 ***
OpenPorchSF    9.452e-02  3.332e-02   2.836 0.004626 **
EnclosedPorch -1.147e-01  3.505e-02  -3.272 0.001092 **
X3SsnPorch     3.450e-02  6.943e-02   0.497 0.619343
ScreenPorch    8.653e-02  3.780e-02   2.289 0.022224 *
PoolArea       1.953e-04  5.382e-02   0.004 0.997105
MiscVal        4.522e-03  4.088e-03   1.106 0.268819
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 76.98 on 1434 degrees of freedom
Multiple R-squared:  0.8159, Adjusted R-squared:  0.813
F-statistic: 276.4 on 23 and 1434 DF,  p-value: < 2.2e-16
```

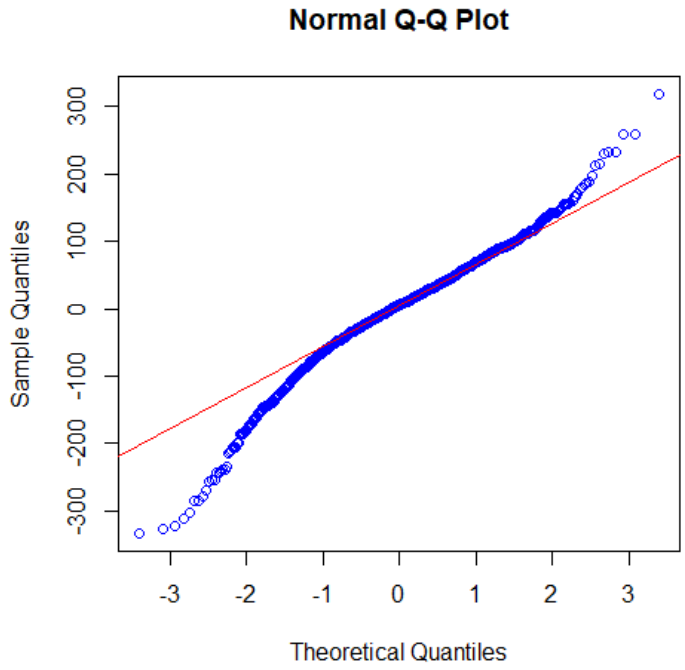
From the summary result, the Multiple R-squared shows 81.59% variability of Housing price can be explained by this model, and the value of Adjusted R-square is 0.813. For the F-statistic, the P-value is less than level of significance 0.01, we can have the statistical evidence that the model is significant.

Then we test the normality and homoscedasticity by normal residuals plot, normal

Q-Q plot and the Kolmogorov-Smirnov test.



Compared with the residual plot before the Box-cox transformation, it is improved at a slight level but still similar as left-opening megaphone, then we attempt to do further statistical test.



As we can see from the Q-Q plot, it seems that the curve improves at a slight level but still does not suppose normality.

To provide more evidences, we use the Kolmogorov-Smirnov test to test whether the data follow the normal distribution. The null hypothesis H0 is the data conform to the theoretical distribution. The result of the KS test is following:

```
    One-sample Kolmogorov-Smirnov test

data:  jitter(residuals_drop_box_cox)
D = 0.062793, p-value = 2.031e-05
alternative hypothesis: two-sided
```

Based on the result, the p-value is smaller than 0.05, the H0 is rejected so that prove the non-normality.

Then we use ncvTest to identify the homoscedasticity:

```
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 1.224009, Df = 1, p = 0.26858
```

Since the P-value is larger than 0.05, we cannot reject the null hypothesis. Therefore, we have the statistical evidence that variance is a constant.

2.3.2. LASSO

A Brief Introduction of LASSO

Multicollinearity often occurs in the high-dimensional dataset. How to eliminate multicollinearity and determine the best model is a key point of regression analysis. The ordinary least squares method is often unsatisfactory in dealing with multicollinearity. There are two main problems in the ordinary least square method: one is the accuracy of prediction; the other is the interpretability of the model. Lasso can handle multicollinearity problem in the dataset. In our previous model 1, we only use correlation coefficients to drop variables directly, which is vague, it may sometimes delete important variables in the final model.

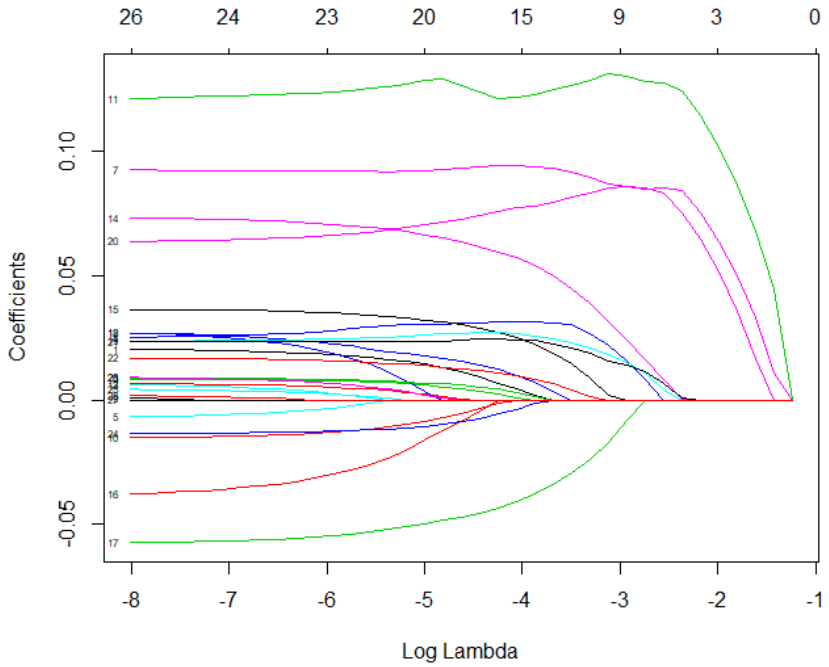
Tibshirani (1996) proposed LASSO (Least Absolute Shrinkage and Selection Operator) algorithm. LASSO is characterized by variable selection and regularization while fitting the general linear model. Variable selection here means that all variables are not put into the model for fitting but are selectively put into the model to obtain better performance of parameters. Regularization is to control the complexity of the model to avoid overfitting.

For linear models, the complexity is directly related to the number of variables in the model. The more variables the model has, the more complete the model is. More variables can often give a seemingly better model, but they also face the risk of overfitting. LASSO adds some penalty term in the objective function.

$$\hat{\beta}^{\text{lasso}} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^{N} \left(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\}.$$

In R, we use glmnet package to call LASSO function directly. First, we draw

coefficients and log(λ) graph to observe the relationship between them intuitively.



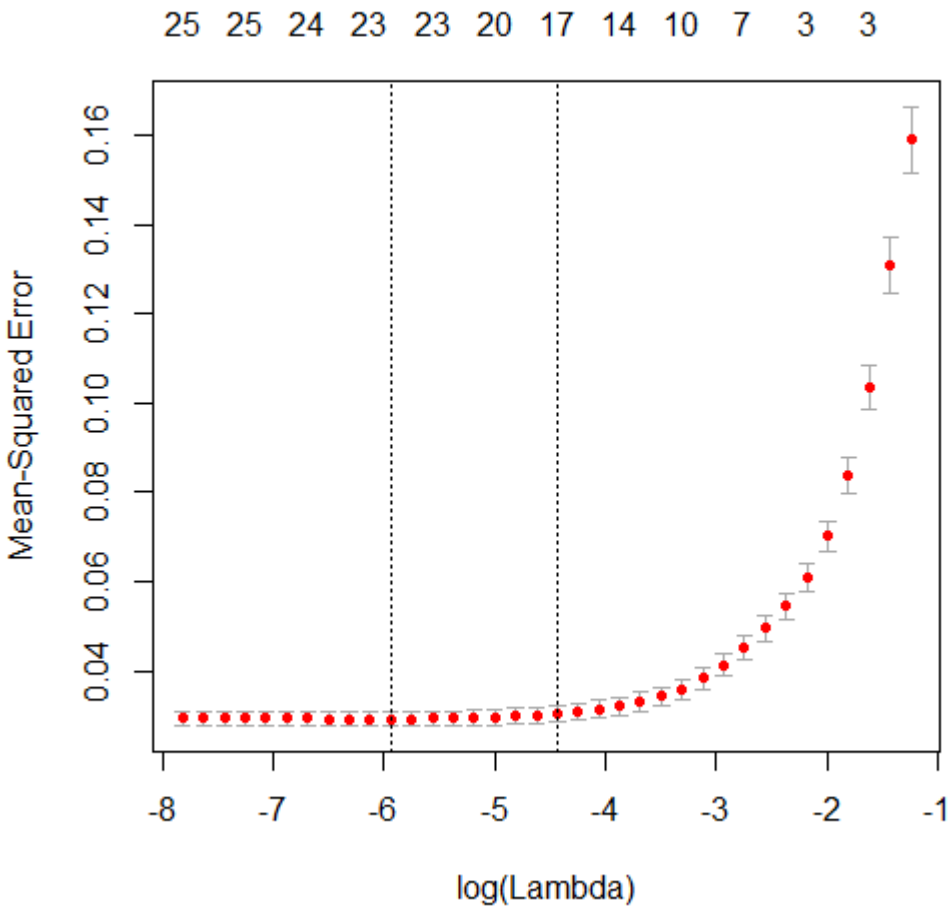
Each curve in the graph represents the trajectory of the coefficients of each independent variable. The Y-axis is the value of the coefficients, the lower X-axis is log(λ), and the upper X-axis is the number of non-zero coefficients in the model. With the increasing of λ, some coefficients are compressed to zero. From above graph, variable 7,11,14, 16,17,20 may be important for modeling.

However, in the above method, all data are used to do a fitting, which is likely to result in over-fitting. At this point, when we use the model to predict a new dataset, the results are likely to be unsatisfactory. Now we will use the cross-validation method to fit and select the model. In R, we directly call cv.glmnet command and print the following result.

```
$name
                mse
"Mean-Squared Error"

$glmnet.fit

Call:  glmnet(x = data_numericial_A_std, y = y, family = "gaussian",      nl$

      Df   %Dev    Lambda
 [1,]  0 0.0000 0.2898000
 [2,]  2 0.1825 0.2401000
 [3,]  3 0.3548 0.1990000
 [4,]  3 0.4782 0.1649000
 [5,]  3 0.5629 0.1366000
 [6,]  3 0.6211 0.1132000
 [7,]  5 0.6612 0.0938100
 [8,]  6 0.6927 0.0777400
 [9,]  7 0.7202 0.0644200
[10,]  8 0.7449 0.0533800
[11,]  9 0.7640 0.0442300
[12,] 10 0.7792 0.0366500
[13,] 10 0.7899 0.0303700
[14,] 11 0.7979 0.0251700
[15,] 14 0.8041 0.0208500
[16,] 15 0.8088 0.0172800
[17,] 16 0.8123 0.0143200
[18,] 17 0.8158 0.0118700
[19,] 18 0.8184 0.0098330
[20,] 19 0.8202 0.0081480
[21,] 20 0.8218 0.0067520
[22,] 20 0.8230 0.0055950
[23,] 22 0.8239 0.0046360
[24,] 23 0.8246 0.0038420
[25,] 23 0.8251 0.0031830
[26,] 23 0.8254 0.0026380
[27,] 23 0.8257 0.0021860
[28,] 24 0.8259 0.0018110
[29,] 24 0.8260 0.0015010
[30,] 24 0.8261 0.0012440
[31,] 24 0.8261 0.0010310
[32,] 24 0.8262 0.0008540
[33,] 25 0.8262 0.0007077
[34,] 25 0.8262 0.0005864
[35,] 25 0.8262 0.0004859
[36,] 25 0.8262 0.0004026
[37,] 26 0.8262 0.0003336

$lambda.min
[1] 0.002637834

$lambda.lse
[1] 0.01186612

attr(,"class")
[1] "cv.glmnet"
```

Each row represents one model. The Df column is the degree of freedom representing the number of non-zero linear coefficients in this model. The %Dev column represents R^2 of the model. The closer it approaches 1, the better the performance of the model will represent. The Lambda column is the corresponding value of λ for each model. As we can see, with the decrease of lambda, more and more independent variables are accepted by the model, and % Dev becomes larger and larger.

At line 25, the model contains 23 variables and % Dev is 0.8251. We can see that when Df is higher than 18, % Dev reaches 0.82. If we continue to reduce λ, namely adding more independent variables to the model, then % Dev cannot be significantly increased.



Now, we plot the cross-validated scatter points. The horizontal axis is log(λ), and the vertical axis is mean square error. The upper and lower bounds of one standard deviation of each point are also drawn. The number at the top of the graph represents the number of non-zero coefficients, and the two vertical dashed lines are the selected lambda after cross-validation. Through cross-validation, we can choose λ with the smallest mean square error or the maximum λ within a standard deviation. Here, we use the first criterion, namely, we use λ with the smallest mean square error to select variables.

We obtain the value of λ with the smallest mean square error is .002637834 and plot all coefficients at this time.

```
> cv.fit$lambda.min
[1] 0.002637834
```

	1
(Intercept)	12.024015156
LotFrontage	0.017729237
LotArea	0.004832256
MasVnrArea	0.007563070
BsmtFinSF1	0.028634701
BsmtFinSF2	-0.002327690
BsmtUnfSF	.
TotalBsmtSF	0.091979817
X1stFlrSF	.
X2ndFlrSF	.
LowQualFinSF	-0.012216352
GrLivArea	0.124609932
BsmtFullBath	0.021818692
BsmtHalfBath	0.001744730
FullBath	0.070213092
HalfBath	0.034667938
BedroomAbvGr	-0.027993778
KitchenAbvGr	-0.053868081
TotRmsAbvGrd	0.016685626
Fireplaces	0.024766971
GarageCars	0.066964661
GarageArea	0.023425594
WoodDeckSF	0.015608433
OpenPorchSF	0.008123232
EnclosedPorch	-0.012201771
X3SsnPorch	0.002021138
ScreenPorch	0.005973212
PoolArea	.
MiscVal	.

From the above result, we can find the coefficients of variables BsmtUnfSF, X1stFlrSF, X2ndFlrSF, PoolArea, and MiscVal become zero.

Next, we will use the remaining variables LotFrontage, LotArea, MasVnrArea, BsmtFinSF1, BsmtFinSF2, TotalBsmtSF, LowQualFinSF, GrLivArea, BsmtFullBath, BsmtHalfBath, FullBath, HalfBath, BedroomAbvGr, KitchenAbvGr, TotRmsAbvGrd, Fireplaces, GarageCars, GarageArea, WoodDeckSF, OpenPorchSF, EnclosedPorch, X3SsnPorch, ScreenPorch to perform future analysis.

After deleting five variables, the model is formed in this way:

Y ~ LotFrontage + LotArea + MasVnrArea + BsmtFinSF1 + BsmtFinSF2 + TotalBsmtSF + LowQualFinSF + GrLivArea + BsmtFullBath + BsmtHalfBath + FullBath + HalfBath + BedroomAbvGr + KitchenAbvGr + TotRmsAbvGrd + Fireplaces + GarageCars + GarageArea + WoodDeckSF + OpenPorchSF + EnclosedPorch + X3SsnPorch + ScreenPorch

The following variables are strong influencial in this model:

FullBath: Full bathrooms above grade.

BsmtFullBath: Basement full bathrooms.

BsmtHalfBath: Basement half bathrooms.

HalfBath: Half baths above grade.

TotRmsAbvGrd: Total rooms above grade (does not include bathrooms).

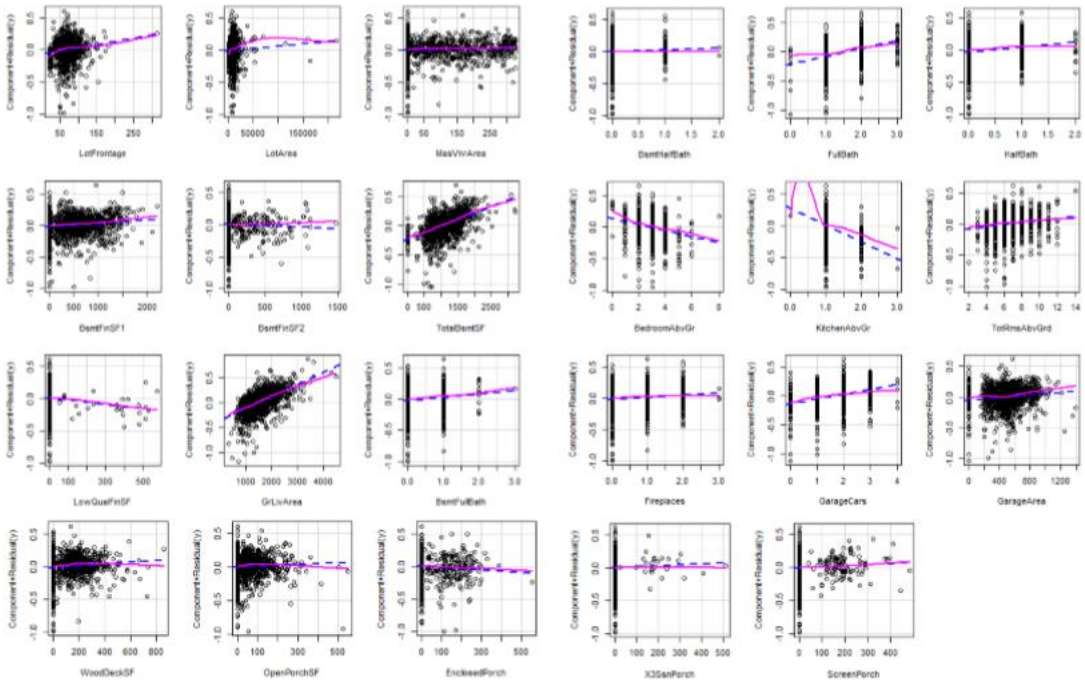
Fireplaces: Number of fireplaces.

GarageCars: Size of garage in car capacity.

```
Call:
lm(formula = y ~ ., data = as.data.frame(data_after_lasso))

Residuals:
     Min       1Q   Median       3Q      Max
-0.97794 -0.07536  0.01747  0.09809  0.61110

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.112e+01  3.000e-02 370.576  < 2e-16 ***
LotFrontage    9.852e-04  2.376e-04   4.147 3.57e-05 ***
LotArea        7.103e-07  4.874e-07   1.457 0.145225
MasVnrArea     8.892e-05  4.985e-05   1.784 0.074693 .
BsmtFinSF1     5.751e-05  1.599e-05   3.597 0.000332 ***
BsmtFinSF2    -4.506e-05  2.964e-05  -1.520 0.128756
TotalBsmtSF    2.226e-04  1.498e-05  14.861  < 2e-16 ***
LowQualFinSF  -3.189e-04  9.490e-05  -3.361 0.000797 ***
GrLivArea      2.387e-04  2.101e-05  11.362  < 2e-16 ***
BsmtFullBath   5.323e-02  1.248e-02   4.265 2.13e-05 ***
BsmtHalfBath   2.827e-02  1.963e-02   1.440 0.149945
FullBath       1.339e-01  1.164e-02  11.500  < 2e-16 ***
HalfBath       7.356e-02  1.091e-02   6.745 2.22e-11 ***
BedroomAbvGr  -4.767e-02  7.850e-03  -6.073 1.61e-09 ***
KitchenAbvGr  -2.617e-01  2.204e-02 -11.875  < 2e-16 ***
TotRmsAbvGrd   1.771e-02  5.899e-03   3.003 0.002719 **
Fireplaces     3.679e-02  8.378e-03   4.392 1.21e-05 ***
GarageCars     8.467e-02  1.369e-02   6.184 8.15e-10 ***
GarageArea     1.111e-04  4.708e-05   2.360 0.018404 *
WoodDeckSF     1.369e-04  3.842e-05   3.562 0.000380 ***
OpenPorchSF    1.394e-04  7.295e-05   1.911 0.056203 .
EnclosedPorch -2.226e-04  7.629e-05  -2.918 0.003583 **
X3SsnPorch     1.567e-04  1.514e-04   1.035 0.301071
ScreenPorch    1.710e-04  8.246e-05   2.073 0.038310 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.168 on 1434 degrees of freedom
Multiple R-squared:  0.8262,    Adjusted R-squared:  0.8234
F-statistic: 296.4 on 23 and 1434 DF,  p-value: < 2.2e-16
```

The result of multiple R-squared is 0.8262. After balancing the fitness and model complexity, adjusted R-squared is 0.8234. The signs of the last column above visualize the level of where t-statistics is significant. Based on the hypothesis is rejected at 0.01 levels. The value is smaller than 0.01 is significant variable and otherwise is not significant variable. As the p-value 2,2e^-16 is smaller than 0.01, the hypothesis is rejected and the β exists.

To learn the relationship between y(SalePrice) and 23 variables, we test the non-linearity in the all 23 regressors. The component-plus-resident plots are following:



The local linear fitting using span = 0.5 and the linear least squares line are plotted in each picture. The span range is small, so we can conclude that all 23 variables are linear.

The criteria donated by the PRESS statistics in this variable selection will be as follows: `[1] 42.13393`

Then we begin our diagnostic analysis.

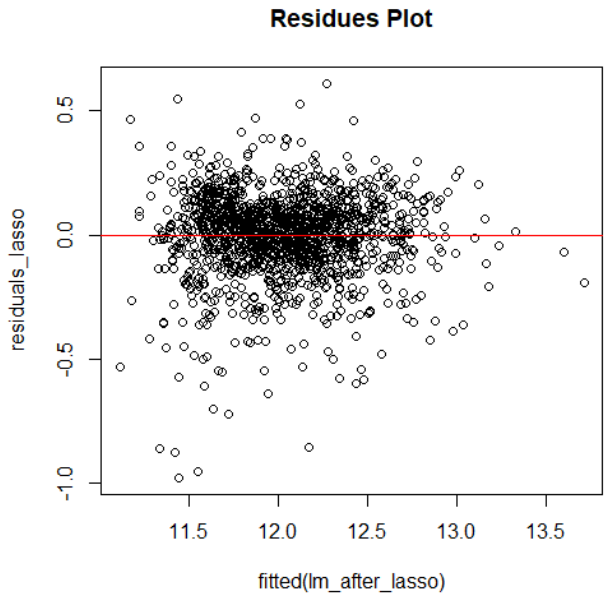
Applying Lasso under delete 5 variables contains BsmtUnfSF, X1stFlrSF, X2ndFlrSF, PoolArea and MiscVal.

The variance inflation factor for the remaining 23 independent variables is following:

LotFrontage	LotArea	MasVnrArea	BsmtFinSF1	BsmtFinSF2
1.290835	1.192394	1.203858	2.474009	1.182545
TotalBsmtSF	LowQualFinSF	GrLivArea	BsmtFullBath	BsmtHalfBath
1.995850	1.101102	5.877695	2.152926	1.135709
FullBath	HalfBath	BedroomAbvGr	KitchenAbvGr	TotRmsAbvGrd
2.117212	1.552508	2.120955	1.219552	4.691810
Fireplaces	GarageCars	GarageArea	WoodDeckSF	OpenPorchSF
1.493971	5.403911	5.157166	1.198047	1.172313
EnclosedPorch	X3SsnPorch	ScreenPorch		
1.124315	1.019455	1.093219		

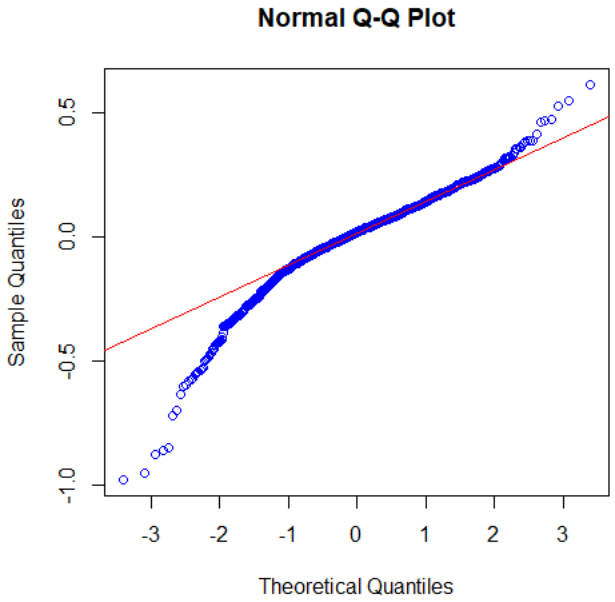
The greater value of VIF implies severer multicollinearity. Normally, If the VIF exceeds 10, the regression model has severe multiple collinearity. Otherwise, it is acceptable to indicate there is no collinearity problem if the results are smaller than 10. In the above results, the value all greater than 0 and up to 5.9. Hence, we can show that after deleting those 5 variables, the remaining variables do not exist severe multiple correlation problem.

Additionally, we test the normality by normal residuals plot and normal Q-Q plot.



The residuals plot is similar as left-opening megaphone. It suggests the variance decreasing with the quality plotted on the x-axis.

The Q-Q plot is following:



The Q-Q plot diagnoses non-normality.

We here use durbinWatsonTest to achieve independent test:

```
> durbinWatsonTest(lm_after_lasso)
 lag Autocorrelation D-W Statistic p-value
   1       0.01195794       1.975796   0.618
 Alternative hypothesis: rho != 0
```

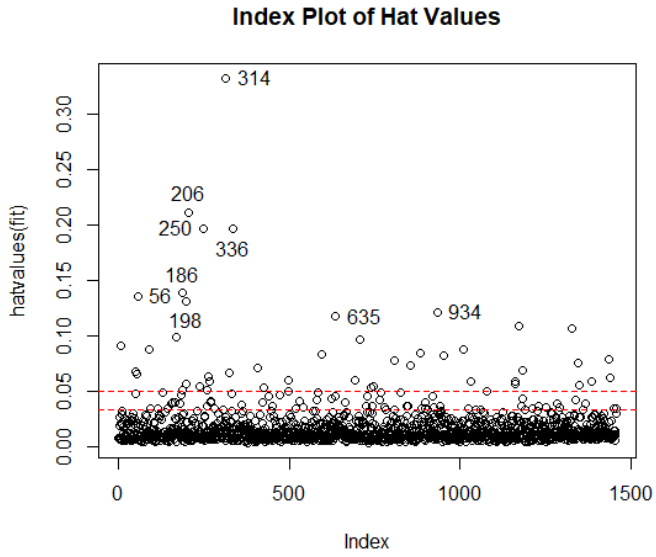
In this case, the p-value is larger than 0.05 which imply the residual independent.

Then we use ncvTest to identify the homoscedasticity:

```
> ncvTest(lm_after_lasso)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 33.40027, Df = 1, p = 7.5014e-09
```

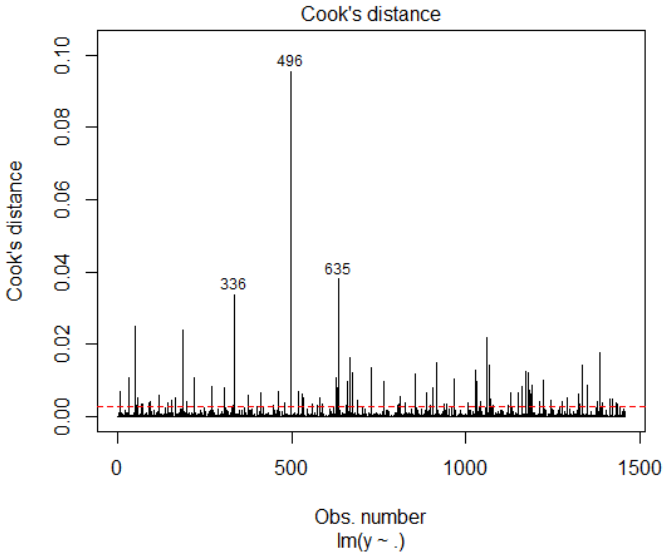
As the hypothesis H0 is residuals variance are same. The P-value is small enough to reject the H0 which indicates the heteroscedasticity, expressing that the random residuals have different variance.

Using the leverage value to detect the model and finding the outliers. The high leverage value is decided by the hat statistics where the hat value is greater two or three times than the mean of hat value.



According to the plot, the red horizontal lines represent twice and three times the average hat value, the 314, 206, 250, 336……points are those high hat-value observations in this plot. We now move on to test the mainly influential points.

As for the influence points diagnosed by Cook's distance. The Cook's distance plot is giving as:



As the plot shown, 495, 635, 336, … are observations with high influential. In some situations, those observations that with high influential feature may potentially exist abnormal influences just like outlier's feature. The Cook's distance plot only identifies the influential points rather than giving reasons. We will compare and find out the outliers and influential points with comprehensive consideration.

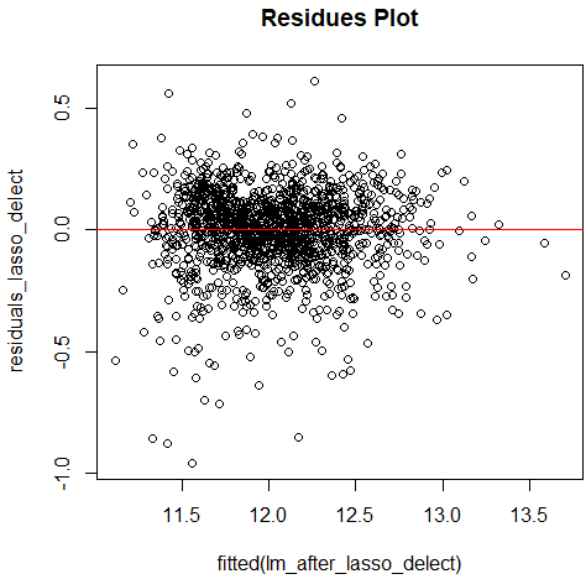
Integrate leverage values and influence points, select points that the hat value and cook distance value above the horizontal reference lines. The intersect points are more likely to be deleted since those points satisfy both conditions and enhance the outlier potentiality. The results will be: `[1]  49  496  635 1385`

After deleting the four observations which considering as outliers above, the result of the new model as following:

```
Call:
lm(formula = y[-lasso_delect, ] ~ ., data = as.data.frame(data_after_lasso_delect))

Residuals:
     Min       1Q   Median       3Q      Max
-0.95884 -0.07461  0.01653  0.09607  0.61419

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.114e+01  2.985e-02 373.130  < 2e-16 ***
LotFrontage    1.042e-03  2.336e-04   4.461 8.80e-06 ***
LotArea        7.357e-07  4.782e-07   1.539 0.124125
MasVnrArea     8.915e-05  4.891e-05   1.823 0.068556 .
BsmtFinSF1     6.232e-05  1.571e-05   3.967 7.64e-05 ***
BsmtFinSF2    -3.916e-05  2.913e-05  -1.344 0.179152
TotalBsmtSF    2.173e-04  1.472e-05  14.758  < 2e-16 ***
LowQualFinSF  -3.896e-04  9.611e-05  -4.054 5.32e-05 ***
GrLivArea      2.313e-04  2.065e-05  11.203  < 2e-16 ***
BsmtFullBath   5.095e-02  1.226e-02   4.154 3.45e-05 ***
BsmtHalfBath   2.604e-02  1.926e-02   1.352 0.176732
FullBath       1.361e-01  1.147e-02  11.861  < 2e-16 ***
HalfBath       7.301e-02  1.071e-02   6.816 1.38e-11 ***
BedroomAbvGr  -4.835e-02  7.771e-03  -6.221 6.47e-10 ***
KitchenAbvGr  -2.809e-01  2.227e-02 -12.615  < 2e-16 ***
TotRmsAbvGrd   1.855e-02  5.803e-03   3.197 0.001420 **
Fireplaces     3.661e-02  8.222e-03   4.453 9.14e-06 ***
GarageCars     8.498e-02  1.345e-02   6.320 3.50e-10 ***
GarageArea     1.070e-04  4.623e-05   2.315 0.020759 *
WoodDeckSF     1.381e-04  3.771e-05   3.661 0.000261 ***
OpenPorchSF    2.342e-04  7.351e-05   3.186 0.001472 **
EnclosedPorch -1.935e-04  7.507e-05  -2.577 0.010065 *
X3SsnPorch     1.589e-04  1.486e-04   1.070 0.284997
ScreenPorch    2.115e-04  8.243e-05   2.565 0.010407 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1648 on 1430 degrees of freedom
Multiple R-squared:  0.8312,    Adjusted R-squared:  0.8285
F-statistic: 306.1 on 23 and 1430 DF,  p-value: < 2.2e-16
```

The result of multiple R-squared changes to 0.8312. After balancing the fitness and model complexity, adjusted R-squared changes to 0.8285. The signs in the last column above visualize the level of where t-statistics is significant. Based on the hypothesis is rejected at 0.01 levels. The value is smaller than 0.01 is significant variable and otherwise is not significant variable. As the p-value same as the previous 2,2e^-16 which is smaller than 0.01, the hypothesis is rejected and the β exists.

We apply the residual plot and ncv-test to homoscedasticity:



The residuals plot is improved a little but still similar as left-opening megaphone. It suggests variance decreasing with the quality plotted on the x-axis.

Then we use ncvTest to identify the homoscedasticity:

```
> ncvTest(lm_after_lasso_delect)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 26.45114, Df = 1, p = 2.7029e-07
```

The P-value is nearly 100 times previous one but still smaller than 0.05 which indicates the heteroscedasticity, expresses the hidden meaning that the random residuals have different variance after deleting outliers.

Above all, the model after deleting outliers still need more improvement procedures.

Due to the non-constant residual variance is diagnosed but the variances are not same, we move to two remedies which are the weighted least squares method and box-cox transformation. In the next step, we apply the weighted least squares. The results are following:

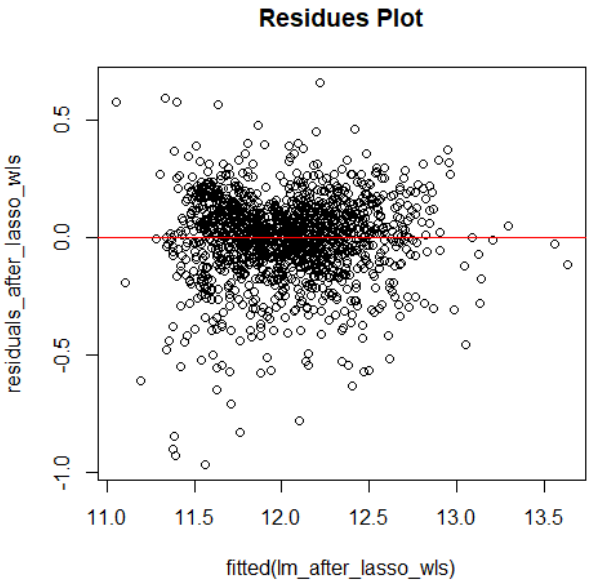
```
Call:
lm(formula = y ~ ., data = as.data.frame(data_drop), weights = 1/abs(residua$

Weighted Residuals:
    Min      1Q  Median      3Q     Max
-2.5203 -0.3075  0.1109  0.3700  1.6902
```

```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.120e+01  1.837e-02 609.958  < 2e-16 ***
LotFrontage    9.071e-04  1.096e-04   8.274 2.93e-16 ***
LotArea        1.107e-06  2.084e-07   5.310 1.27e-07 ***
MasVnrArea     1.139e-04  2.177e-05   5.233 1.92e-07 ***
BsmtFinSF2    -5.856e-05  1.619e-05  -3.616 0.000309 ***
BsmtUnfSF      2.105e-05  6.499e-06   3.239 0.001226 **
X1stFlrSF      4.758e-04  1.034e-05  46.008  < 2e-16 ***
X2ndFlrSF      2.913e-04  7.596e-06  38.348  < 2e-16 ***
LowQualFinSF  -5.248e-05  5.664e-05  -0.927 0.354267
BsmtFullBath   1.059e-01  5.552e-03  19.082  < 2e-16 ***
BsmtHalfBath   5.454e-02  1.065e-02   5.122 3.44e-07 ***
FullBath       1.317e-01  4.785e-03  27.513  < 2e-16 ***
HalfBath       7.169e-02  5.675e-03  12.634  < 2e-16 ***
BedroomAbvGr  -5.088e-02  3.728e-03 -13.648  < 2e-16 ***
KitchenAbvGr  -3.024e-01  1.446e-02 -20.911  < 2e-16 ***
Fireplaces     2.542e-02  3.598e-03   7.063 2.53e-12 ***
GarageCars     1.268e-01  3.926e-03  32.312  < 2e-16 ***
WoodDeckSF     1.502e-04  1.878e-05   7.996 2.62e-15 ***
OpenPorchSF    1.203e-04  2.981e-05   4.035 5.75e-05 ***
EnclosedPorch -1.229e-04  3.874e-05  -3.171 0.001550 **
X3SsnPorch     1.314e-04  1.079e-04   1.218 0.223439
ScreenPorch    2.152e-04  4.190e-05   5.135 3.20e-07 ***
PoolArea      -3.056e-05  1.254e-04  -0.244 0.807464
MiscVal       -1.723e-05  6.771e-06  -2.545 0.011036 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4585 on 1434 degrees of freedom
Multiple R-squared:  0.9692,    Adjusted R-squared:  0.9687
F-statistic:  1961 on 23 and 1434 DF,  p-value: < 2.2e-16
```

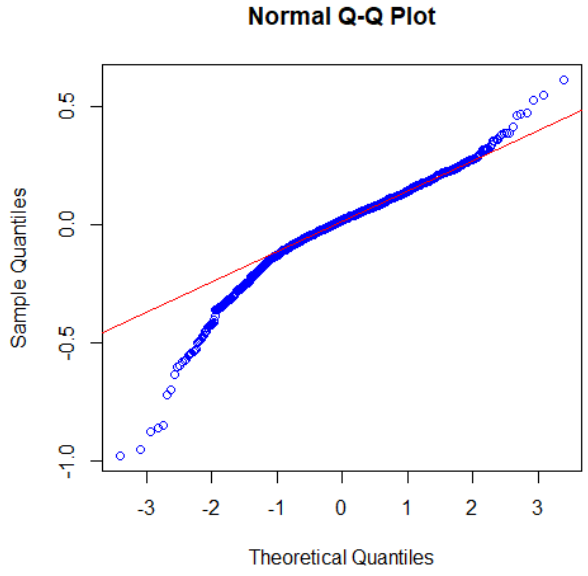
The multiple R-squared is improved from 0.8262 to 0.9692. The same improvement happened in adjusted R-squared. The signs in the last column above visualize the level of where t-statistics is significant. Based on the hypothesis is rejected at 0.01 levels. The significant variables become more. As the p-value still is 2,2e^-16 which is smaller than 0.01, the hypothesis is rejected and the β exists.

The criteria donated by the PRESS statistics in this variable selection will be as follows: `[1] 46.59305`

Additionally, we test the normality and homoscedasticity by normal residuals plot, normal Q-Q plot and the Kolmogorov-Smirnov test.



The residuals plot is improved on a slight level but still similar as left-opening megaphone. It suggests variance decreasing with the quality plotted on the x-axis.



The Q-Q plot diagnoses non-normality. But the area overlaps to the reference line becomes better.

We here use the Kolmogorov-Smirnov test to test whether the data follow the normal distribution. Its hypothesis H0 is the data conform to the theoretical distribution. The result of the KS test is following:

```
        One-sample Kolmogorov-Smirnov test

data:  jitter(residuals_lasso)
D = 0.081558, p-value = 7.538e-09
alternative hypothesis: two-sided
```

According to the result, the p-value is smaller than 0.05 that prove the non-

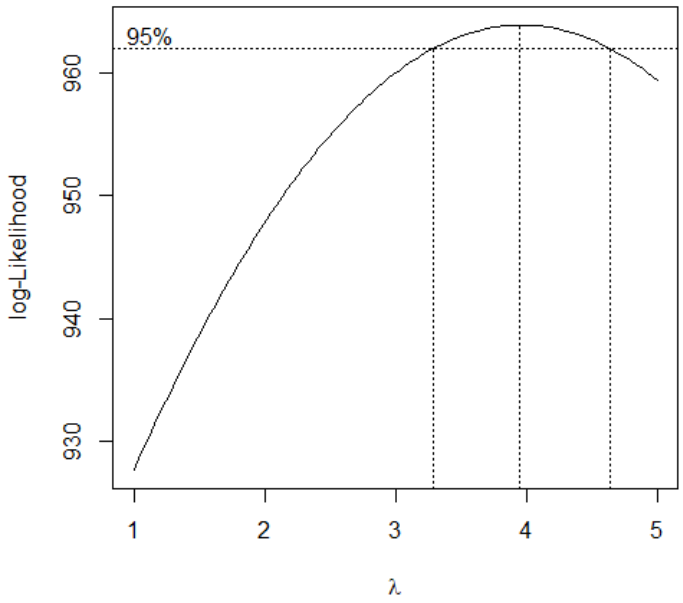
normality.

Then we use ncvTest to identify the homoscedasticity:

```
> ncvTest(lm_after_lasso_wls)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 0.09864287, Df = 1, p = 0.75346
```

The P-value is larger than 0.05 which indicate the homoscedasticity, which expresses that the random residuals have same variance. The weighted least square method is effective. However, call back to the previous residuals plot, the weights may still need to change considering the approximate left-opening megaphone shape.

When the non-constant variances are diagnosed, we contemplate another remedy to make the residual variances more nearly equal which is box-cox transformation. The transformation starts from the determination of the λ based on the log maximum likelihood function. The plot shows the procedure and the final λ are follows:




```
> powerTransform(lm_after_lasso)$lambda
      Y1
3.968731
```

And here the Box-Cox transformation is defined as:

$$y(\lambda) = \begin{cases} \dfrac{y^{3.968731} - 1}{3.968731}, \lambda \neq 0 \\ \ln y, \quad \lambda = 0 \end{cases}$$

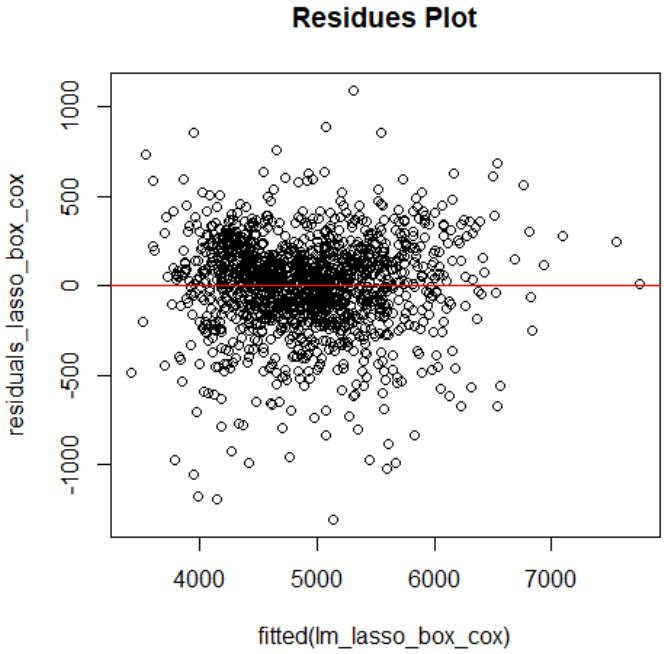
The brief results after transformation as following:

```
Call:
lm(formula = y_lasso_box_cox ~ ., data = as.data.frame(data_after_lasso))

Residuals:
     Min       1Q   Median       3Q      Max
-1306.95  -127.11    24.74   155.95  1091.26
```

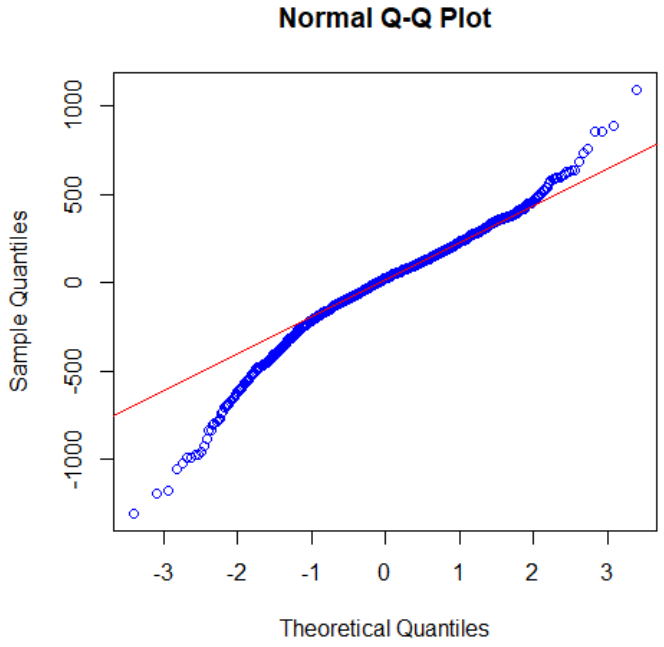
```
Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    3.444e+03  4.699e+01  73.300  < 2e-16 ***
LotFrontage    1.684e+00  3.721e-01   4.526 6.50e-06 ***
LotArea        1.434e-03  7.634e-04   1.878 0.060524 .
MasVnrArea     1.745e-01  7.809e-02   2.234 0.025617 *
BsmtFinSF1     1.075e-01  2.504e-02   4.294 1.87e-05 ***
BsmtFinSF2    -8.782e-02  4.643e-02  -1.891 0.058763 .
TotalBsmtSF    3.631e-01  2.346e-02  15.474  < 2e-16 ***
LowQualFinSF  -5.594e-01  1.486e-01  -3.764 0.000174 ***
GrLivArea      4.360e-01  3.290e-02  13.251  < 2e-16 ***
BsmtFullBath   7.520e+01  1.955e+01   3.848 0.000124 ***
BsmtHalfBath   3.235e+01  3.074e+01   1.052 0.292803
FullBath       2.024e+02  1.824e+01  11.096  < 2e-16 ***
HalfBath       1.053e+02  1.708e+01   6.165 9.16e-10 ***
BedroomAbvGr  -9.983e+01  1.230e+01  -8.119 1.00e-15 ***
KitchenAbvGr  -4.313e+02  3.452e+01 -12.495  < 2e-16 ***
TotRmsAbvGrd   3.311e+01  9.239e+00   3.584 0.000350 ***
Fireplaces     4.943e+01  1.312e+01   3.767 0.000172 ***
GarageCars     1.264e+02  2.145e+01   5.896 4.65e-09 ***
GarageArea     1.801e-01  7.375e-02   2.442 0.014738 *
WoodDeckSF     2.187e-01  6.018e-02   3.635 0.000288 ***
OpenPorchSF    2.593e-01  1.143e-01   2.270 0.023380 *
EnclosedPorch -3.578e-01  1.195e-01  -2.995 0.002796 **
X3SsnPorch     1.898e-01  2.372e-01   0.800 0.423644
ScreenPorch    2.545e-01  1.292e-01   1.970 0.048981 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 263.1 on 1434 degrees of freedom
Multiple R-squared:  0.8395,    Adjusted R-squared:  0.8369
F-statistic: 326.1 on 23 and 1434 DF,  p-value: < 2.2e-16
```

The result of multiple R-squared is 0.8395. After balancing the fitness and model complexity, adjusted R-squared is 0.8369. The signs in the last column above visualize the level of where t-statistics is significant. Based on the hypothesis is rejected at 0.01 levels. The value is smaller than 0.01 is significant variable and otherwise is not significant variable. As the p-value 2,2e^-16 is smaller than 0.01, the hypothesis is rejected and the β exists.

Move on to next step, we test the normality and homoscedasticity by normal residuals plot, normal Q-Q plot and the Kolmogorov-Smirnov test.



Comparing with the situation before transformation, the residuals plot is improved at a slight level but still similar as left-opening megaphone. It suggests variance decreasing with the quality plotted on the x-axis.



The Q-Q plot diagnoses non-normality. But the area overlaps to the reference line is improved a little.

To provide more evidences, we use the Kolmogorov-Smirnov test to test whether the data follow the normal distribution. Its' hypothesis H0 is the data conform to the theoretical distribution. The result of the KS test is following:

```
        One-sample Kolmogorov-Smirnov test

data:  jitter(residuals_lasso_box_cox)
D = 0.065552, p-value = 7.233e-06
alternative hypothesis: two-sided
```

Based on the result, the p-value is smaller than 0.05, the H0 is rejected so that prove the non-normality.

Then we use ncvTest to identify the homoscedasticity:

```
> ncvTest(lm_lasso_box_cox)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 1.777689, Df = 1, p = 0.18243
```

The P-value is larger than 0.05 which indicate the homoscedasticity, which expresses that the random residuals have same variance.

3. Conclusion and Discussion.

3.1 Conclusion

	Model	R^2	$R^2{}_{adj}$	s^2	PRESS	P-value (NCV test)	P-value (Independent test)
Model 1 y ~ LotFrontage + ⋯+ MiscVal	Drop highly correlated variables	0.8058	0.8026	0.0316	47.37008	$2.892e^{-7}$	0.9
	Drop potential outliers	0.8127	0.8097	0.0302	44.77641	$1.5855e^{-7}$	0.736
	Weighted least squares transformation	0.9851	0.9848	0.1230	45.60279	0.11962	0.848
	Box-cox transformation	0.8159	0.813	-	-	0.2688	0.786
Model 2 y ~ LotFrontage +⋯+ ScreenPorch	Drop	0.8262	0.8234	0.0266	42.1339	$7.5014e^{-9}$	0.618
	Drop potential outliers	0.8312	0.8285	0.0272	40.2943	$2.7029e^{-7}$	0.9
	Weighted least squares transformation	0.9692	0.9687	0.2102	46.59305	0.75346	0.71
	Box-cox transformation	0.8395	0.8396	-	-	0.18243	0.658

To summarize, we list several test statistics as the criteria to make the model comparison. Multiple R-squared consider the comparison of fitness, and Adjusted R-squared combine the fitness and model complexity. S-squared is also called residual mean square. PRESS gives the level of the prediction residual. NCV test determines the heteroscedasticity, while Independent test tests the independence of the residual.

It is ideal if there exists a model with larger Multiple R-squared and Adjusted R-squared, lower s-squared and PRESS. As well as the one can pass the NCV test and Independent test.

For the Box-cox transformation, since we have done the transformation on the response variable, the standard of the response variable has been changed. Therefore,

we do not consider making the comparison in s-square and PRESS.

Due to adding weight in the weighted least squares model, it will directly cause a significant enhancement on R-squared and Adjusted R-squared. Therefore, we cannot directly make the comparison in R-squared and Adjusted R-squared based on the weighted least squares model.

All things considered, we suggest the Model 2 with dropping potential outliers since it has the smallest PRESS, a smaller s-squared, and acceptable Multiple R-squared and Adjusted R-squared. If we can know the expert's suggestions, we can consider the weighted least squares method.

3.2 Discussion

In this section, we propose six improvements that can be further explored in the future.

3.2.1 The parameter λ in LASSO

In our previous model, we used the lambda value at the minimum MSE as the criterion for selecting variables. Now, we can use lambda within a standard deviation and the least parameters in the model to be our criterion for selecting variables. If we use this criterion, we will get a model of 17 variables.

The models of these 17 variables can be further analyzed to obtain the model with the least parameters and the highest prediction accuracy.

```
> cv.fit$lambda.1se
[1] 0.01186612
```

We will use lambda.1se in our model, and we will obtain the following result.

```
> coefficients
29 x 1 sparse Matrix of class "dgCMatrix"
                          1
(Intercept)    12.024015156
LotFrontage     0.009313445
LotArea         .
MasVnrArea      0.003393260
BsmtFinSF1      0.031163177
BsmtFinSF2      .
BsmtUnfSF       .
TotalBsmtSF     0.093775455
X1stFlrSF       .
X2ndFlrSF       .
LowQualFinSF   -0.003470620
GrLivArea       0.123922615
BsmtFullBath    0.014254502
BsmtHalfBath    .
FullBath        0.061685038
HalfBath        0.028978887
BedroomAbvGr   -0.004175166
KitchenAbvGr   -0.045356891
TotRmsAbvGrd    .
Fireplaces      0.027263195
GarageCars      0.074101345
GarageArea      0.024404380
WoodDeckSF      0.011959323
OpenPorchSF     0.005217753
EnclosedPorch  -0.007327658
X3SsnPorch      .
ScreenPorch     .
PoolArea        .
MiscVal         .
```

```
Call:
lm(formula = y ~ ., data = as.data.frame(data_after_lasso_lse))

Residuals:
     Min       1Q   Median       3Q      Max
-0.98462 -0.07704  0.01828  0.10094  0.63076

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    1.113e+01  2.982e-02 373.318  < 2e-16 ***
LotFrontage    1.068e-03  2.339e-04   4.565 5.42e-06 ***
MasVnrArea     9.461e-05  4.982e-05   1.899 0.057778 .
BsmtFinSF1     6.675e-05  1.495e-05   4.466 8.58e-06 ***
TotalBsmtSF    2.197e-04  1.480e-05  14.848  < 2e-16 ***
LowQualFinSF  -3.102e-04  9.527e-05  -3.256 0.001157 **
GrLivArea      2.772e-04  1.737e-05  15.963  < 2e-16 ***
BsmtFullBath   4.466e-02  1.157e-02   3.859 0.000119 ***
FullBath       1.312e-01  1.159e-02  11.316  < 2e-16 ***
HalfBath       7.216e-02  1.092e-02   6.605 5.57e-11 ***
BedroomAbvGr  -3.600e-02  7.011e-03  -5.135 3.20e-07 ***
KitchenAbvGr  -2.463e-01  2.137e-02 -11.526  < 2e-16 ***
Fireplaces     4.105e-02  8.241e-03   4.982 7.07e-07 ***
GarageCars     8.801e-02  1.371e-02   6.418 1.86e-10 ***
GarageArea     1.070e-04  4.727e-05   2.263 0.023807 *
WoodDeckSF     1.292e-04  3.791e-05   3.410 0.000669 ***
OpenPorchSF    1.432e-04  7.321e-05   1.956 0.050663 .
EnclosedPorch -2.540e-04  7.606e-05  -3.339 0.000864 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1687 on 1440 degrees of freedom
Multiple R-squared:  0.8239,    Adjusted R-squared:  0.8218
F-statistic: 396.3 on 17 and 1440 DF,  p-value: < 2.2e-16
```

As we can see, R^2 is 0.8239, and R^2_{adj} is 0.8218. By our calculation, we obtain PRESS = 42.32429 in our model.

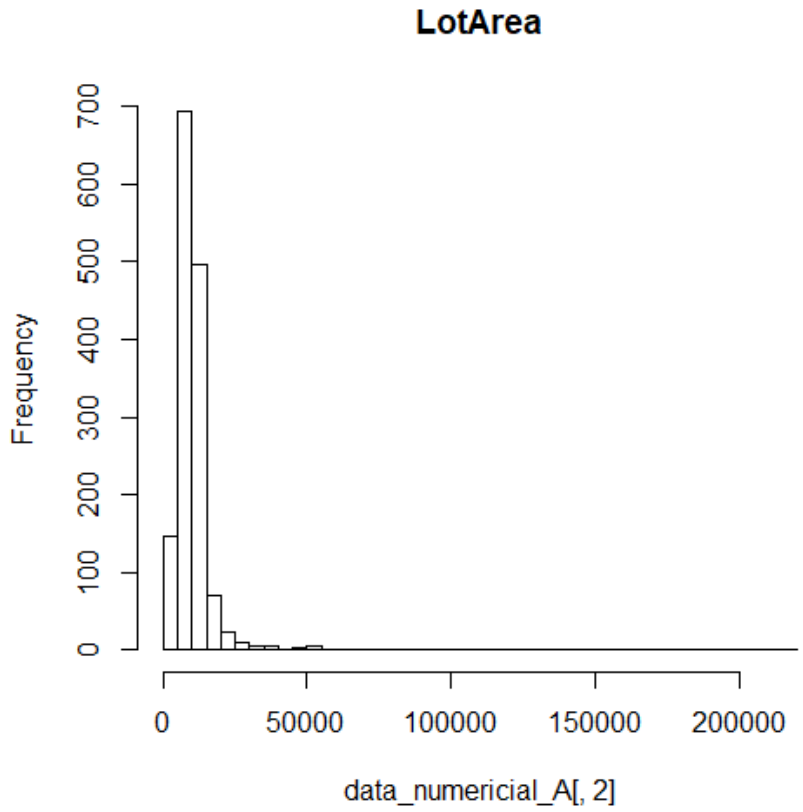
3.2.2 Independent variables transformation

The skewness of some independent variables is too high. We can use some transformation such as $\log(x+1)$ and x^a to reduce the skewness of independent variables. The technique may play an essential role in stabilizing the variance of residuals.

We compute skewness and kurtosis of all numerical variables in R and obtain the following result.

```
> Skewness_dataframe <- data.frame(Skewness,Variable=name)
> Skewness_dataframe
     Skewness      Variable
1   1.7173179   LotFrontage
2  12.5480649       LotArea
3   1.3917795    MasVnrArea
4   0.7632165    BsmtFinSF1
5   4.2431805    BsmtFinSF2
6   0.9190089     BsmtUnfSF
7   0.5106509   TotalBsmtSF
8   0.8858110     X1stFlrSF
9   0.8112855     X2ndFlrSF
10  8.9864348  LowQualFinSF
11  1.0089124     GrLivArea
12  0.5891443  BsmtFullBath
13  4.0916815  BsmtHalfBath
14  0.0312065      FullBath
15  0.6786523      HalfBath
16  0.2118885  BedroomAbvGr
17  4.4756590  KitchenAbvGr
18  0.6591433  TotRmsAbvGrd
19  0.6307596    Fireplaces
20 -0.3416728    GarageCars
21  0.1314770    GarageArea
22  1.5426260    WoodDeckSF
23  2.3350164   OpenPorchSF
24  3.0808146 EnclosedPorch
25 10.2759286    X3SsnPorch
26  4.1104574   ScreenPorch
27 15.9161436      PoolArea
28 24.4097786       MiscVal
> Kurtosis_dataframe <- data.frame(Kurtosis,Variable=name)
> Kurtosis_dataframe
      Kurtosis      Variable
1   14.6180805   LotFrontage
2  212.5587963       LotArea
3    0.5736805    MasVnrArea
4   -0.1197811    BsmtFinSF1
5   19.9758308    BsmtFinSF2
6    0.4633808     BsmtUnfSF
7    1.7534010   TotalBsmtSF
8    1.1038095     X1stFlrSF
9   -0.5625728     X2ndFlrSF
10  82.7079810  LowQualFinSF
11   2.0453534     GrLivArea
12  -0.8682907  BsmtFullBath
13  16.2811313  BsmtHalfBath
14  -0.8740187      FullBath
15  -1.0732254      HalfBath
16   2.2054106  BedroomAbvGr
17  21.3854401  KitchenAbvGr
18   0.8452477  TotRmsAbvGrd
19  -0.2976664    Fireplaces
20   0.2128231    GarageCars
21   0.7494242    GarageArea
22   2.9833030    WoodDeckSF
23   8.4274732   OpenPorchSF
24  10.3529293 EnclosedPorch
25 122.8875081    X3SsnPorch
26  18.3115045   ScreenPorch
27 256.5052028      PoolArea
28 696.6820029       MiscVal
```

For example, if we plot the histogram of LotArea variable, we will observe that there is a long tail in it, which means that high skewness occurs in the variable.



3.2.3 The weights of Weighted Least Square

Our previous choice of weights for Weighted Least Square was based on the reciprocal of the absolute residual value. In the future, we can consider more diverse weights.

3.2.4 Merging variables and creating new variables

In the original variables, we find that some variables have similar meanings, so we can merge the variables before analyzing the data. In the future, we may create some new variables to represent some essential features, which can better reflect the characteristics of data.

For example, we find three variables have similar meanings. 1stFlrSF represents first-floor square feet, 2ndFlrSF represents second-floor square feet, and TotalSF represents total square feet in the original data set. All those three variables describe the square feet of the house, so we can merge those three variables into one new variable called total square feet of the house. Thus, we can define the TotalSF variable as the sum of those three variables in the future analysis.

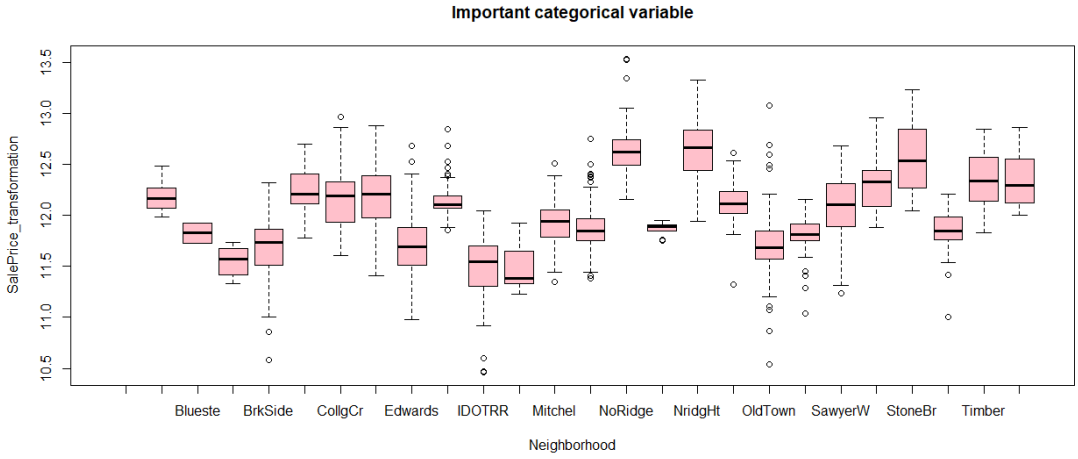
3.2.5 Adding categorical variables

In our previous model, we only considered all numerical variables, and then we could consider adding categorical variables. For unordered categorical variables, we can introduce dummy variables. For ordered categorical variables, the existing model will

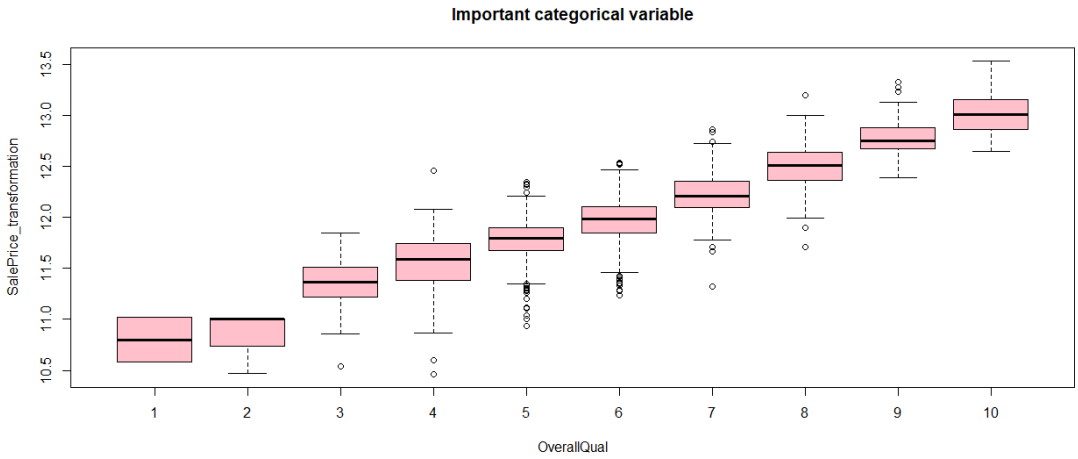
not be satisfied, so we need to use a new model.

We selected some important categorical variables and drew their box plots. Meanwhile, we observed that these variables have a significant impact on the final price prediction.

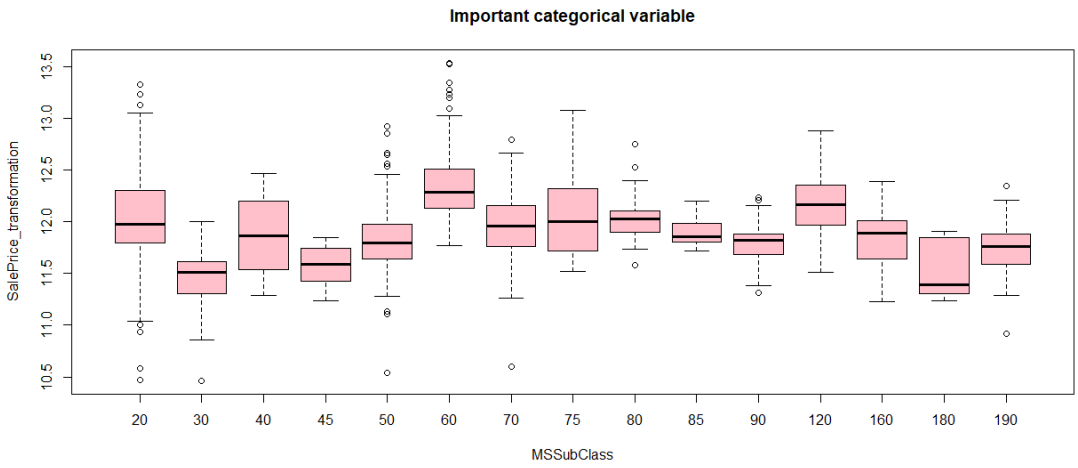
With different neighborhood, housing prices fluctuate considerably.



With the increase of the score of overall quality, the house price is increasing.



With different category of MSSubClass, house prices have changed a lot.



3.2.6 Using more advanced methods

In the future, we can use more advanced methods such as XGBoost and LightGBM and use the weighted average of the predicted values of each method as the final predicted values.

```
> hist(data_numericial_A[,27],breaks=100)
> hist(data_numericial_A[,2],breaks=100)
> hist(data_numericial_A[,2],breaks=50)
> data_after_lasso <- data_numericial_A[,c(-6,-8,-9,-27,-28)]
> lm_after_lasso <- lm(y~.,data=as.data.frame(data_after_lasso))
> data_drop <- data_numericial_A[,c(-4,-7,-11,-18,-21)]
> lm_drop <- lm(y~.,data=as.data.frame(data_drop))
> anova(lm_drop,lm_after_lasso)
Analysis of Variance Table

Model 1: y ~ LotFrontage + LotArea + MasVnrArea + BsmtFinSF2 + BsmtUnfSF +
    X1stFlrSF + X2ndFlrSF + LowQualFinSF + BsmtFullBath + BsmtHalfBath +
    FullBath + HalfBath + BedroomAbvGr + KitchenAbvGr + Fireplaces +
    GarageCars + WoodDeckSF + OpenPorchSF + EnclosedPorch + X3SsnPorch +
    ScreenPorch + PoolArea + MiscVal
Model 2: y ~ LotFrontage + LotArea + MasVnrArea + BsmtFinSF1 + BsmtFinSF2 +
    TotalBsmtSF + LowQualFinSF + GrLivArea + BsmtFullBath + BsmtHalfBath +
    FullBath + HalfBath + BedroomAbvGr + KitchenAbvGr + TotRmsAbvGrd +
    Fireplaces + GarageCars + GarageArea + WoodDeckSF + OpenPorchSF +
    EnclosedPorch + X3SsnPorch + ScreenPorch
  Res.Df    RSS Df Sum of Sq F Pr(>F)
1   1434 45.215
2   1434 40.456  0    4.7597


> data_after_lasso_lse <- data_numericial_A[,c(-2,-5,-6,-8,-9,-13,-18,-25,-26,-27,-$
> lm_after_lasso_lse <- lm(y~.,data=as.data.frame(data_after_lasso_lse))
> anova(lm_after_lasso_lse,lm_after_lasso)
Analysis of Variance Table

Model 1: y ~ LotFrontage + MasVnrArea + BsmtFinSF1 + TotalBsmtSF + LowQualFinSF +
    GrLivArea + BsmtFullBath + FullBath + HalfBath + BedroomAbvGr +
    KitchenAbvGr + Fireplaces + GarageCars + GarageArea + WoodDeckSF +
    OpenPorchSF + EnclosedPorch
Model 2: y ~ LotFrontage + LotArea + MasVnrArea + BsmtFinSF1 + BsmtFinSF2 +
    TotalBsmtSF + LowQualFinSF + GrLivArea + BsmtFullBath + BsmtHalfBath +
    FullBath + HalfBath + BedroomAbvGr + KitchenAbvGr + TotRmsAbvGrd +
    Fireplaces + GarageCars + GarageArea + WoodDeckSF + OpenPorchSF +
    EnclosedPorch + X3SsnPorch + ScreenPorch
  Res.Df    RSS Df Sum of Sq     F   Pr(>F)
1   1440 40.997
2   1434 40.456  6   0.54149 3.199 0.004009 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Reference

Robert I. (2016). *R in Action Data Analysis and Graphics with R (second edition)*. Beijing: Posts&Telecom Press

Raymond H. (2005). *Classical and Modern Regression with Applications (second edition)*. Beijing: Higher Education Press

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*. *Series B (Methodological)*, 267-288.

Appendix

```
library(corrplot)
library(MASS)
library(carData)
library(car)
library(ridge)
library(Matrix)
library(foreach)
library(glmnet)

###          download          dataset          ONLY          train
data!!!!#####################################
#####################          file          =          train.csv
#####################################
#####  rm(list = ls())  ####
data <- read.csv(file.choose(), stringsAsFactors = FALSE, header=
TRUE )

### look the dimension and structure of the dataset
dim(data)
str(data)

#####################################################################
############
########################   non   informative   outlier   dectetion
###################################
plot(data$GrLivArea, data$SalePrice, main = "With Outliers")

## remove the outliers non-informative
data <- data[-which(train$GrLivArea > 4000 & train$SalePrice <
3e+05),]
#### data <- data

### checking again
plot(data$GrLivArea, data$SalePrice, main = "With Outliers")
#####################################################################
############


#####################################################################
#############
#####################                                    transfirmation
########################################
```

```
#### first we explore the distrbution of the SalePrice
#### test the normality of the SalePrice
hist(data$SalePrice, breaks = 50, main = "Right skewed
distribution")
qqnorm(data$SalePrice,col='blue', main = 'Non-transformation Q-Q
Plot')
qqline(data$SalePrice,col='red')

###try to use log(x+1) trsformation
data$SalePrice <- log(data$SalePrice + 1)
## data <- data
hist(data$SalePrice, breaks = 50, main = "After transformation")
qqnorm(data$SalePrice,col='blue', main = 'After transformation Q-Q
Plot')
qqline(data$SalePrice,col='red')
#####################################################################
##############


#####################################################################
########################################
#####################################################################
########################################
################################       missing    data    analysis
#################################################

################## visualize missing data ###################
Missing_Ratio <- sort(colSums(is.na(data))/nrow(data),decreasing =
TRUE)

Missing_Ratio <- Missing_Ratio[Missing_Ratio!=0]
print(Missing_Ratio)

############                        visailize                        the
ratio#############################################################
#############################################
barCenters         <-         barplot(Missing_Ratio,col         =
rainbow(length(Missing_Ratio)),main='Missing     Rate',axes     =
FALSE,axisnames=FALSE,border ="white")
text(x = barCenters, y = par("usr")[3]-.2, srt = 60, adj = 1, labels
= names(Missing_Ratio), xpd = TRUE)

barCenters  <-  barplot(Missing_Ratio,col  =  "lightblue",axes  =
FALSE,horiz=TRUE,border ="white")
```

```
text(y = barCenters, x = par("usr")[3],adj = 1, labels =
names(Missing_Ratio), xpd = TRUE)
axis(1, labels = seq(0,9,by=1), at = seq(0,9,by=1), las = 1 , col
= "gray")
##################################################################
##################################################################
#######

#################### Imputing missing data ################

###### categorical data

data$PoolQC[is.na(data$PoolQC)] <- 'None'
data$MiscFeature[is.na(data$MiscFeature)] <-'None'
data$Alley[is.na(data$Alley)] <-'None'
data$Fence[is.na(data$Fence)] <-'None'
data$FireplaceQu[is.na(data$FireplaceQu)] <-'None'
data$GarageType[is.na(data$GarageType)] <-'None'
data$GarageFinish[is.na(data$GarageFinish)] <-'None'
data$GarageQual[is.na(data$GarageQual)] <-'None'
data$GarageCond[is.na(data$GarageCond)] <-'None'

###### numercial data
data$GarageYrBlt[is.na(data$GarageYrBlt)] <- 0
data$GarageArea[is.na(data$GarageFinishGarageArea)] <- 0
data$GarageCars [is.na(data$GarageCars )] <- 0
data$BsmtFinSF2[is.na(data$BsmtFinSF2)] <- 0
data$BsmtFinSF1[is.na(data$BsmtFinSF1)] <- 0
data$BsmtUnfSF[is.na(data$BsmtUnfSF)] <- 0
data$TotalBsmtSF[is.na(data$TotalBsmtSF)] <- 0
data$BsmtFullBath[is.na(data$BsmtFullBath)] <- 0
data$BsmtHalfBath[is.na(data$BsmtHalfBath)] <- 0
#########################################

######### categorical data
data$BsmtQual[is.na(data$BsmtQual)] <-'None'
data$BsmtCond[is.na(data$BsmtCond)] <-'None'
data$BsmtExposure[is.na(data$BsmtExposure)] <-'None'
data$BsmtFinType1[is.na(data$BsmtFinType1)] <-'None'
data$BsmtFinType2[is.na(data$BsmtFinType2)] <-'None'
data$MasVnrArea[is.na(data$MasVnrArea)] <-'None'
data$MasVnrType[is.na(data$MasVnrType)] <-'None'

data$MSZoning[is.na(data$MSZoning)] <-'RL'
```

```
data$Functional[is.na(data$Functional)] <-'Typ'
data$Electrical[is.na(data$Electrical)] <-'Sbrkr'
data$KitchenQual[is.na(data$KitchenQual)] <-'TA'

### max(table(data$Exterior2nd))
data$Exterior1st[is.na(data$Exterior1st)] <- 'VinylSd'
data$Exterior2nd[is.na(data$Exterior2nd)] <-' VinylSd'
data$SaleType[is.na(data$SaleType)] <-'WD'
data$MSSubClass[is.na(data$MSSubClass)] <-'None'
############################################################


###### numercial data
##LotFrontage, usind median
#sum(is.na(data$LotFrontage))
#!is.na(data$LotFrontage)

data$MasVnrArea[data$MasVnrArea==NA]                              <-
median(as.numeric(data$MasVnrArea!=NA))##########################
################
data$MasVnrArea[data$MasVnrArea=='None']                          <-
median(as.numeric(data$MasVnrArea!='None'))

#sum(is.na(data$LotFrontage))

data$LotFrontage <- as.numeric(data$LotFrontage)
data$LotFrontage[is.na(data$LotFrontage)]                         <-
median(data$LotFrontage[!is.na(data$LotFrontage)])

####double check the missing rata

Missing_Ratio <- sort(colSums(is.na(data))/nrow(data),decreasing =
TRUE)

Missing_Ratio <- Missing_Ratio[Missing_Ratio!=0]
print(Missing_Ratio)

###export clean data
####

write.table(data,    file    ="C:\\Users\\user\\Desktop\\Regression
Analysis (1002) (Dr. Hua-Jun YE)\\Group_Project\\train_clean.csv",
sep =",", row.names =FALSE)
############################################################
```

```
##########################################
#########################################################################
##########################################
#########################################################################
##########################################


#####################################        simply        explore
muticollinearity ####################################

library(corrplot)
data$MasVnrArea                                                        <-
as.numeric(data$MasVnrArea)#######################################
####3
data$LotFrontage <- as.numeric(data$LotFrontage)

correlations                                                           <-
cor(data[,c(4,5,27,35,37,38,39,c(44:53),55,57,62,63,c(67:72),76)],
use="everything")
corrplot(correlations, method="circle", type="lower",  sig.level =
0.01, insig = "blank")
#########################################################################
########################################
corrplot(correlations, method = "color", outline = T, cl.pos = 'n',
rect.col = "black",  tl.col = "indianred4", addCoef.col = "black",
number.digits = 2, number.cex = 0.60, tl.cex = 0.7, cl.cex = 1, col
= colorRampPalette(c("green4","white","red"))(100))

#############################################################

#################  TAKE CARE !!!!! ############################
data <- data[,-1]  ### delect the id column
#############################################################

########### using transformation data
data$MasVnrArea                                                        <-
as.numeric(data$MasVnrArea)#######################################
####3
data$LotFrontage <- as.numeric(data$LotFrontage)

lm <- lm(data$SalePrice~.,data=data)
summary(lm)

##############  (9 not defined because of singularities)
```

```
#####################################################################
###################################
#####################################################################
###################################
######################### prices without transformation file =
train_clean_+without transformation ###
#########################   the   importance   of   transformation
#########################################
#####################################################################
###################################
data_without_transformation <- read.csv(file.choose(),head=T)

price_without_transformation <- data_without_transformation[,82]

data_without_transformation   <-   data_without_transformation[,-
c(1,81,82)]
data_without_transformation$MasVnrArea                          <-
as.numeric(data_without_transformation$MasVnrArea)
lm_without          <-          lm(price_without_transformation~.,
data=data_without_transformation)
summary(lm_without)

#####################################################################
###################################
#####################################################################
###################################
#####################################################################
###################################
###########trial and error

plot(fitted(lm_without),summary(lm_without)$residuals,main='The
Residual   Plot',xlab='Predicted   Value',ylab='Ordinary   Prices
Residuals')
qqnorm(summary(lm_without)$residuals,col='blue')
qqline(summary(lm_without)$residuals,col='red')

library(car)
vif(lm)

durbinWatsonTest(lm)
ncvTest(lm)

#####################################################################
#####################################
```

```
#####################################################################
#####################################
library(MASS) ###trial and error

data_clean <- read.csv(file.choose(),header=T)###train_clean
SalePrice <- data_clean[,81]
data_clean <- data_clean[,-1]  ### delect the id column
data_clean <- data_clean[,-80]  ### delect the saleprice

ridgelm <- lm.ridge(SalePrice~.,data=data_clean)
summary(ridgelm)
print(ridgelm)
plot(lm.ridge(SalePrice~.,data=data_clean,lambda                  =
seq(0,0.0001,0.0001)))
select(lm.ridge(SalePrice~.,data=data_clean,lambda                =
seq(0,0.0001,0.0001)))
#####################################################################
#########################################
#####################################################################
#########################################
library(ridge) ###trial and error

mod <- linearRidge(SalePrice~.,data=data_clean)
summary(mod)

#stepAIC(lm, direction = “forward”)
step(lm,direction = c("forward"))
#####################################################################
#########################################
#####################################################################
#########################################


#####################################################################
################
############### first we only consider numercial data in our data
set
##################

data_L <- read.csv(file.choose(),header=T)## train_clean_FINAL

##c(4,5,27,35,37,38,39,c(44:53),55,57,62,63,c(67:72),76)
#data_numericial                                                  <-
as.matrix(data_L[,c(4,5,27,35,37,38,39,c(44:53),55,57,62,63,c(67:
```

```
72),76)])

data_numericial                                                    <-
apply(as.matrix(unlist(data_L[,c(4,5,27,35,37,38,39,c(44:53),55,5
7,62,63,c(67:72),76)])), 2, as.numeric)
data_numericial_A <- matrix(data_numericial,ncol=28)

y <- as.matrix(data_L[,81])
######beta <- solve(t(X)%*%X) %*% t(X) %*% y

name                                                               <-
c('LotFrontage','LotArea','MasVnrArea','BsmtFinSF1','BsmtFinSF2',
'BsmtUnfSF','TotalBsmtSF','X1stFlrSF','X2ndFlrSF','LowQualFinSF',
'GrLivArea','BsmtFullBath','BsmtHalfBath','FullBath','HalfBath','
BedroomAbvGr','KitchenAbvGr','TotRmsAbvGrd','Fireplaces','GarageC
ars','GarageArea','WoodDeckSF','OpenPorchSF','EnclosedPorch','X3S
snPorch','ScreenPorch','PoolArea','MiscVal')

data_frame_lm <- data.frame(data_numericial_A)
colnames(data_numericial_A) <- name
colnames(data_frame_lm) <- name
###################################!!!!!!!!!!!!!!!!!!!!!!!!!!

##data matrix
X <- cbind(1,data_numericial_A)
#data_numericial_A_std <-scale(data_numericial_A)

###
### using conditional number to chech the multicollinearity
kappa(t(X)%*%X)#[1] 2.026486e+19

solve(t(X)%*%X)

lm <- lm(y~.,data=data_frame_lm)
summary(lm)

##vif
library(car)
vif(lm)### CANNOT BE CALCULATED!!!!!

##in the previous lm model, we find NA in it , NA in lm will be
instable in our future analysis
##so we need to drop some high correlation variablb
## using our corr matrix to drop out directly
```

```
##dropVars  <-  c('YearRemodAdd',  'GarageYrBlt',  'GarageArea',
'GarageCond', 'TotalBsmtSF', 'TotalRmsAbvGrd', 'BsmtFinSF1')

library(carData)
library(car)
library(MASS)

######################## Model 1 ##################
data_drop <- data_numericial_A[,c(-4,-7,-11,-18,-21)]
lm_drop <- lm(y~.,data=as.data.frame(data_drop))
summary(lm_drop)

##########################
X_1 <- cbind(1,data_drop)
kappa(t(X_1)%*%X_1)#9757115663

##vif_drop using coor
vif(lm_drop)

###PRESS drop
press_drop        <-        sum((residuals(lm_drop)/(1        -
lm.influence(lm_drop)$hat))^2)
press_drop #47.37008

residuals_drop <- resid(lm_drop)
#studentized_residuals                                          <-
summary(lm_drop)$residuals/(summary(lm_drop)$sigma*sqrt(1-
lm.influence(lm_drop)$hat))
plot(fitted(lm_drop),residuals_drop,main='Residual Plot')
abline(h=0,col='red')

###Test the norimality assumption
qqnorm(residuals_drop,col='blue')
qqline(residuals_drop,col='red')

ks.test(jitter(residuals_drop),pnorm,mean(x),sd(x))##??,mean(x),s
d(x)??
####### the function of jitter() is to avoid the same data

shapiro.test(residuals_drop)

##independent test
durbinWatsonTest(lm_drop) # p = 0.886
```

```
##same distribution
ncvTest(lm_drop) # p = 5.6489e-08?  2.892e-07

##linearity
crPlots(lm_drop)

##Regression Influence Plot
#influence.plot(lm_drop)
influencePlot(lm_drop,main="InfluencePlot",sub="circle   size   is
proportional to cook's distance")
#influence.measures(lm_drop)##??

cutoff <- 4/(length(y)-length(lm_drop$coefficients)-2)
plot(lm_drop,which=4,cook.levels=cutoff)
abline(h=cutoff,lty=2,col="red")


### high leverage point
hat.plot<-function(fit){
p<-length(coefficients(fit))
n<-length(fitted(fit))
plot(hatvalues(fit),main="Index Plot of Hat Values")
abline(h=c(2,3)*p/n,col="red",lty=2)
identify(1:n,hatvalues(fit),names(hatvalues(fit)))
}
hat.plot(lm_drop)

###hii
p <- length(coefficients(lm_drop))
n <- length(fitted(lm_drop))
drop_delect_hii <- which(hatvalues(lm_drop)>=3*p/n)
###ti
drop_delect_ti <- which(abs(rstudent(lm_drop))>2)
###delect #[1]   49  496  635  854 1031 1385
drop_delect <- intersect(drop_delect_hii,drop_delect_ti)


##
#####delect data_drop[c(49,496,635,854,1031,1385),]
drop_delect <- c(49,496,635,854,1031,1385)
data_after_drop_delect <- data_drop[-drop_delect,]
lm_after_drop_delect                    <-                    lm(y[-
drop_delect,]~.,data=as.data.frame(data_after_drop_delect))
summary(lm_after_drop_delect)
```

```
residuals_drop_delect <- resid(lm_after_drop_delect)
plot(fitted(lm_after_drop_delect),residuals_drop_delect,main='Res
idual Plot')
abline(h=0,col='red')

#qqnorm(residuals_drop_delect)
#qqline(residuals_drop_delect)
ncvTest(lm_after_drop_delect)


####if                              we                              use
outlierTest(lm_drop)###################################### ????
#####delect data_drop[c(31,496,916,968,812,632),]
drop_delect_out <- c(31,496,916,968,812,632)
data_after_drop_delect_out <- data_drop[-drop_delect_out,]
lm_after_drop_delect_out                 <-                 lm(y[-
drop_delect_out,]~.,data=as.data.frame(data_after_drop_delect_out
))
summary(lm_after_drop_delect_out)

residuals_drop_delect_out <- resid(lm_after_drop_delect_out)
plot(fitted(lm_after_drop_delect_out),residuals_drop_delect_out,m
ain='Residual Plot')
abline(h=0,col='red')

#qqnorm(residuals_drop_delect_out)
#qqline(residuals_drop_delect_out)
ncvTest(lm_after_drop_delect_out)

outlierTest(lm_after_drop_delect_out)
#########################################################################
#####


######## box cox transformation for model 1
library(carData)
library(car)

powerTransform(lm_drop)$lambda
#boxcox(lm_drop)$lambda

boxcox(lm_drop, lambda=seq(1, 5, by=0.1))
```

```
y_drop_box_cox        <-        (y^powerTransform(lm_drop)$lambda-
1)/powerTransform(lm_drop)$lambda

lm_drop_box_cox                                                    <-
lm(y_drop_box_cox~.,data=as.data.frame(data_drop))
summary(lm_drop_box_cox)

residuals_drop_box_cox <- resid(lm_drop_box_cox)
plot(fitted(lm_drop_box_cox),residuals_drop_box_cox,main='Residua
l Plot')
abline(h=0,col='red')

#studentized_residuals_box_cox                                     <-
summary(lm_drop_box_cox)$residuals/(summary(lm_drop_box_cox)$sigm
a*sqrt(1-lm.influence(lm_drop_box_cox)$hat))
#plot(fitted(lm_drop_box_cox),studentized_residuals_box_cox,main=
'Studentized Residual Plot')
#abline(h=0,col='red')

#Test the norimality assumption
qqnorm(residuals_drop_box_cox,col='blue')
qqline(residuals_drop_box_cox,col='red')

ks.test(jitter(residuals_drop_box_cox),pnorm,mean(residuals_drop_
box_cox),sd(residuals_drop_box_cox))
shapiro.test(residuals_drop_box_cox)

######## Box-Tidwell transformation for model 1

##boxTidwell(y~.,data=as.data.frame(data_drop+runif(1,0,1)))
##box.tidwell(yy~.,data=as.data.frame(data_drop))

############################################################
#########    weighted least square WLS for model 1
############################################################

residuals_drop <- resid(lm_drop)
lm_drop_wls                                                        <-
lm(y~.,weights=1/abs(residuals_drop),data=as.data.frame(data_drop
))
summary(lm_drop_wls)#R2 0.9851

###PRESS drop wls
press_drop_wls      <-      sum((residuals(lm_drop_wls)/(1      -
```

```
lm.influence(lm_drop_wls)$hat))^2)
press_drop_wls #45.60279

residuals_drop_wls <- resid(lm_drop_wls)
qqnorm(residuals_drop_wls ,col='blue')
qqline(residuals_drop_wls ,col='red')
ks.test(jitter(residuals_drop_wls),pnorm,mean(residuals_drop_wls)
,sd(residuals_drop_wls))

ncvTest(lm_drop_wls)# p = 0.11962

residuals_drop_wls <- resid(lm_drop_wls)
plot(fitted(lm_drop_wls),residuals_drop_wls,main='Residual Plot')
abline(h=0,col='red')

####################### Model 2 ####################
################## using Lasso, we delect 6,8,9,27,28
data_after_lasso <- data_numericial_A[,c(-6,-8,-9,-27,-28)]
lm_after_lasso <- lm(y~.,data=as.data.frame(data_after_lasso))
summary(lm_after_lasso)

#########################
X_2 <- cbind(1,data_after_lasso)
kappa(t(X_2)%*%X_2)#7946982753


##vif using lasso
vif(lm_after_lasso)

###PRESS LASSO
press_lasso      <-      sum((residuals(lm_after_lasso)/(1      -
lm.influence(lm_after_lasso)$hat))^2)
press_lasso #42.13393

residuals_lasso <- resid(lm_after_lasso)
plot(fitted(lm_after_lasso),residuals_lasso,main='Residual Plot')
abline(h=0,col='red')

###Test the norimality assumption
qqnorm(residuals_lasso,col='blue')
qqline(residuals_lasso,col='red')
#plot(rstudent(press_lasso))

ks.test(jitter(residuals_lasso),pnorm,mean(x),sd(x))##??,mean(x),
```

```
sd(x)??
####### the function of jitter() is to avoid the same data

shapiro.test(residuals_lasso)

##independent test
durbinWatsonTest(lm_after_lasso) # p = 0.66

##same distribution
ncvTest(lm_after_lasso) # p = 7.5014e-09

##linearity
crPlots(lm_after_lasso)

#step(lm_after_lasso)

##Regression Influence Plot
#influence.plot(lm_after_lasso)
influencePlot(lm_after_lasso,main="InfluencePlot",sub="circle size
is proportional to cook's distance")
#influence.measures(lm_drop)

cutoff <- 4/(length(y)-length(lm_after_lasso$coefficients)-2)
plot(lm_after_lasso,which=4,cook.levels=cutoff)
abline(h=cutoff,lty=2,col="red")


### high leverage point
hat.plot<-function(fit){
p<-length(coefficients(fit))
n<-length(fitted(fit))
plot(hatvalues(fit),main="Index Plot of Hat Values")
abline(h=c(2,3)*p/n,col="red",lty=2)
identify(1:n,hatvalues(fit),names(hatvalues(fit)))
}
hat.plot(lm_after_lasso)

###hii
p <- length(coefficients(lm_after_lasso))
n <- length(fitted(lm_after_lasso))
lasso_delect_hii <- which(hatvalues(lm_after_lasso)>=3*p/n)
###ti
lasso_delect_ti <- which(abs(rstudent(lm_after_lasso))>2)
###delect #[1]   49  496  635 1385
```

```
lasso_delect <- intersect(lasso_delect_hii,lasso_delect_ti)


#####delect data_after_lasso[c(49,496,635,1385),]
lasso_delect <- c(49,496,635,1385)
data_after_lasso_delect <- data_after_lasso[-lasso_delect,]
lm_after_lasso_delect                  <-                  lm(y[-
lasso_delect,]~.,data=as.data.frame(data_after_lasso_delect))
summary(lm_after_lasso_delect)

residuals_lasso_delect <- resid(lm_after_lasso_delect)
plot(fitted(lm_after_lasso_delect),residuals_lasso_delect,main='R
esidual Plot')
abline(h=0,col='red')

#qqnorm(residuals_lasso_delect)
#qqline(residuals_lasso_delect)
ncvTest(lm_after_lasso_delect)


######## box cox transformation for model 2
library(carData)
library(car)

powerTransform(lm_after_lasso)$lambda
#boxcox(lm_after_lasso)$lambda

boxcox(lm_after_lasso, lambda=seq(1, 5, by=0.1))

y_lasso_box_cox    <-    (y^powerTransform(lm_after_lasso)$lambda-
1)/powerTransform(lm_after_lasso)$lambda

lm_lasso_box_cox                                                <-
lm(y_lasso_box_cox~.,data=as.data.frame(data_after_lasso))
summary(lm_lasso_box_cox)

residuals_lasso_box_cox <- resid(lm_lasso_box_cox)
plot(fitted(lm_lasso_box_cox),residuals_lasso_box_cox,main='Resid
ual Plot')
abline(h=0,col='red')

#Test the norimality assumption
qqnorm(residuals_lasso_box_cox,col='blue')
qqline(residuals_lasso_box_cox,col='red')
```

```
#hist(residuals_lasso_box_cox,breaks=100)

ks.test(jitter(residuals_lasso_box_cox),pnorm,mean(x),sd(x))

press_lasso     <-     sum((residuals(lm_lasso_box_cox)/(1     -
lm.influence(lm_lasso_box_cox)$hat))^2)
press_lasso #103567021


############################################################
#########    weighted least square WLS for model 2
############################################################

residuals_lasso <- resid(lm_after_lasso)
lm_after_lasso_wls                                          <-
lm(y~.,weights=1/abs(residuals_lasso),data=as.data.frame(data_dro
p))
summary(lm_after_lasso_wls)

###PRESS after lasso wls
press_after_lasso_wls  <-  sum((residuals(lm_after_lasso_wls)/(1  -
lm.influence(lm_after_lasso_wls)$hat))^2)
press_after_lasso_wls #46.59305

qqnorm(residuals_lasso,col='blue')
qqline(residuals_lasso,col='red')
ks.test(jitter(residuals_lasso),pnorm,mean(residuals_lasso),sd(re
siduals_lasso))

ncvTest(lm_after_lasso_wls)

residuals_after_lasso_wls <- resid(lm_after_lasso_wls)
plot(fitted(lm_after_lasso_wls),residuals_after_lasso_wls,main='R
esidual Plot')
abline(h=0,col='red')

############
library(MASS)
stepAIC(lm_after_lasso,direction="backward")

#################################################################
#####################################################
#################################################################
#####################################################
```

```
#####################################################################
#########################################################
#####################################################################
#########################################################


################Ridge regression ########################
#########################################
anova(lm_after_lasso,lm_drop)###???????
#####################################################################
############################

###Influence case  (DFFITS, DFBETA, Cook distance)
influence.measures(lm)

#####################################################################
#########################

library(Matrix)
library(foreach)
library(glmnet)

#########################################################standerlized
the data, -mu /sd

##### how to use LASSO to reduce parameter
#####  LASSO additional meterial

#write.table(data_numericial,                                    file
="C:\\Users\\user\\Desktop\\Regression Analysis (1002) (Dr. Hua-
Jun YE)\\Group_Project\\112.csv", sep =",", row.names =FALSE)

data_numericial_A_std <- scale(data_numericial_A)
cv.fit <- cv.glmnet(data_numericial_A_std, y, family="gaussian",
nlambda=50, alpha=1,standardize=TRUE)#standardize=TRUE?

names(cv.fit)
print(cv.fit)

cv.fit$lambda.min # [1] 0.002637834

coefficients_1se <- coef(cv.fit,s=cv.fit$lambda.1se)##put the part
into discussion
coefficients_1se
```

```
coefficients_min <- coef(cv.fit,s=cv.fit$lambda.min)
coefficients_min

active_index<-which(coefficients!=0)
active.coefficients<-coefficients[Active.Index]
active_index

plot(cv.fit)

#####??? the different between cv.glmnet and glmnet
fit   <-   glmnet(data_numericial_A_std,   y,   family="gaussian",
nlambda=50, alpha=1,standardize=TRUE)
plot(fit,xvar="lambda", label=TRUE)
print(fit)
coefficients<-coef(fit,s=fit$lambda.min)

coefficients
##plot(fit)

####ridge? ##alpha = 0 ## ??
fit_ridge <- glmnet(data_numericial_A_std, y, family="gaussian",
nlambda=50, alpha=0,standardize=TRUE)
plot(fit_ridge,xvar="lambda", label=TRUE)

##############                                              Discussion,1
################################################-- add new method??
########################       Lasso       using       lambda.1se
######################
################### using Lasso, we delect -2,-5,-6,-8,-9,-13,-18,-
25,-26,-27,-28

library(carData)
library(car)

data_after_lasso_1se <- data_numericial_A[,c(-2,-5,-6,-8,-9,-13,-
18,-25,-26,-27,-28)]
lm_after_lasso_1se                                                <-
lm(y~.,data=as.data.frame(data_after_lasso_1se))
summary(lm_after_lasso_1se)

#########################
X_1se <- cbind(1,data_after_lasso_1se)
kappa(t(X_1se)%*%X_1se)#73168856
```

```
##vif using lasso_1se
vif(lm_after_lasso_1se)

###PRESS LASSO
press_lasso_1se    <-    sum((residuals(lm_after_lasso_1se)/(1    -
lm.influence(lm_after_lasso_1se)$hat))^2)
press_lasso_1se

residuals_lasso_1se <- resid(lm_after_lasso_1se)
plot(fitted(lm_after_lasso_1se),residuals_lasso_1se,main='Residue
s Plot')
abline(h=0,col='red')

###Test the norimality assumption
qqnorm(residuals_lasso_1se,col='blue')
qqline(residuals_lasso_1se,col='red')
#plot(rstudent(press_lasso_1se))

ks.test(jitter(residuals_lasso_1se),pnorm,mean(x),sd(x))##??,mean
(x),sd(x)??
####### the function of jitter() is to avoid the same data

shapiro.test(residuals_lasso_1se)


##independent test
durbinWatsonTest(lm_after_lasso_1se) # p = 0.66

##same distribution
ncvTest(lm_after_lasso_1se) # p = 7.5014e-09

##linearity
crPlots(lm_after_lasso_1se)

############# Discussion,2##############################

library(e1071)
Kurtosis <- NULL
Skewness <- NULL
for (i in c(1:dim(data_numericial_A)[2])){
Kurtosis <- c(Kurtosis,kurtosis(data_numericial_A[,i]))
Skewness <- c(Skewness,skewness(data_numericial_A[,i]))
```

```
}


Skewness_dataframe <- data.frame(Skewness,Variable=name)
Skewness_dataframe

Kurtosis_dataframe <- data.frame(Kurtosis,Variable=name)
Kurtosis_dataframe

hist(data_numericial_A[,2],breaks=50,main='LotArea')
#hist(data_numericial_A[,25],breaks=50,main='X3SsnPorch')

#Skewness_sort <- sort(Skewness,decreasing = TRUE)
#Skewness_sort

############# Discussion,5##############################
##train_clean_+without transformation
library(MASS)
boxplot(CL~sex,data=crabs,col='pink',xlab='sex',ylab='CL',main='c
rabs')

data_5 <- read.csv(file.choose(),header=T)

boxplot(data_5[,81]~data_5$Neighborhood,col='pink',xlab='Neighbor
hood',ylab='SalePrice_transformation',main='Important  categorical
variable')
boxplot(data_5[,81]~data_5$MSSubClass,col='pink',xlab='MSSubClass
',ylab='SalePrice_transformation',main='Important     categorical
variable')
boxplot(data_5[,81]~data_5$OverallQual,col='pink',xlab='OverallQu
al',ylab='SalePrice_transformation',main='Important   categorical
variable')
```