MTH404 R Project

Jiqing Li

2023-04-20

Table of Contents

Data

We collected the data from ``kaggle datasets" named as "KC_Housesales_Data". The link of the data: https://www.kaggle.com/swathiachath/kc-housesales-data

Online property companies offer valuations of houses using machine learning techniques. The aim of this report is to predict the house sales in King County, Washington State, USA using Multiple Linear Regression (MLR). The dataset consisted of historic data of houses sold between May 2014 to May 2015.

```
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyve
rse 2.0.0 ──
## ✔ dplyr     1.1.1     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_co
nflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to
force all conflicts to become errors

library(corrplot)

## corrplot 0.92 loaded

library(lubridate)
library(readr)
library(caTools)

library(GGally)

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2

library(caret)

## 载入需要的程辑包：lattice
##
## 载入程辑包：'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
library(leaps)

library(dplyr)
library(ggplot2)
library(gridExtra)

##
## 载入程辑包：'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

Load the data

By reading the provided train data in Excel, I have select some major columns from it as our traindata in R Project.

```r
traindata <- read.csv("~/train1.csv", header=TRUE)
testdata <- read.csv("~/test.csv", header=TRUE)

str(traindata)

## 'data.frame':    1460 obs. of  22 variables:
##  $ Id          : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ SalePrice   : int  208500 181500 223500 140000 250000 143000 3070
00 200000 129900 118000 ...
##  $ LotArea     : int  8450 9600 11250 9550 14260 14115 10084 10382 6
120 7420 ...
##  $ Neighborhood: chr  "CollgCr" "Veenker" "CollgCr" "Crawfor" ...
##  $ 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 1
939 ...
##  $ YearRemodAdd: int  2003 1976 2002 1970 2000 1995 2005 1973 1950 1
950 ...
##  $ TotalBsmtSF : int  856 1262 920 756 1145 796 1686 1107 952 991 ...
##  $ X1stFlrSF   : int  856 1262 920 961 1145 796 1694 1107 1022 1077
 ...
##  $ X2ndFlrSF   : int  854 0 866 756 1053 566 0 983 752 0 ...
##  $ GrLivArea   : int  1710 1262 1786 1717 2198 1362 1694 2090 1774 1
077 ...
##  $ BsmtFullBath: int  1 0 1 1 1 1 1 1 0 1 ...
##  $ BsmtHalfBath: int  0 1 0 0 0 0 0 0 0 0 ...
##  $ FullBath    : int  2 2 2 1 2 1 2 2 2 1 ...
##  $ HalfBath    : int  1 0 1 0 1 1 0 1 0 0 ...
##  $ BedroomAbvGr: int  3 3 3 3 4 1 3 3 2 2 ...
##  $ KitchenAbvGr: int  1 1 1 1 1 1 1 1 2 2 ...
##  $ GarageCars  : int  2 2 2 3 3 2 2 2 2 1 ...
```

```
##  $ GarageArea   : int  548 460 608 642 836 480 636 484 468 205 ...
##  $ MoSold       : int  2 5 9 2 12 10 8 11 4 1 ...
##  $ YrSold       : int  2008 2007 2008 2006 2008 2009 2007 2009 2008 2
008 ...
```

```
head(traindata,10)
```

```
##    Id SalePrice LotArea Neighborhood OverallQual OverallCond YearBui
lt
## 1   1    208500    8450      CollgCr           7           5      20
03
## 2   2    181500    9600      Veenker           6           8      19
76
## 3   3    223500   11250      CollgCr           7           5      20
01
## 4   4    140000    9550      Crawfor           7           5      19
15
## 5   5    250000   14260      NoRidge           8           5      20
00
## 6   6    143000   14115      Mitchel           5           5      19
93
## 7   7    307000   10084      Somerst           8           5      20
04
## 8   8    200000   10382       NWAmes           7           6      19
73
## 9   9    129900    6120      OldTown           7           5      19
31
## 10 10    118000    7420      BrkSide           5           6      19
39
##    YearRemodAdd TotalBsmtSF X1stFlrSF X2ndFlrSF GrLivArea BsmtFullBa
th
## 1          2003         856       856       854      1710
 1
## 2          1976        1262      1262         0      1262
 0
## 3          2002         920       920       866      1786
 1
## 4          1970         756       961       756      1717
 1
## 5          2000        1145      1145      1053      2198
 1
## 6          1995         796       796       566      1362
 1
## 7          2005        1686      1694         0      1694
 1
## 8          1973        1107      1107       983      2090
 1
## 9          1950         952      1022       752      1774
 0
## 10         1950         991      1077         0      1077
```

```
 1
##    BsmtHalfBath FullBath HalfBath BedroomAbvGr KitchenAbvGr GarageCa
rs
## 1             0        2        1            3            1
 2
## 2             1        2        0            3            1
 2
## 3             0        2        1            3            1
 2
## 4             0        1        0            3            1
 3
## 5             0        2        1            4            1
 3
## 6             0        1        1            1            1
 2
## 7             0        2        0            3            1
 2
## 8             0        2        1            3            1
 2
## 9             0        2        0            2            2
 2
## 10            0        1        0            2            2
 1
##    GarageArea MoSold YrSold
## 1         548      2   2008
## 2         460      5   2007
## 3         608      9   2008
## 4         642      2   2006
## 5         836     12   2008
## 6         480     10   2009
## 7         636      8   2007
## 8         484     11   2009
## 9         468      4   2008
## 10        205      1   2008
```

Clean and Modify Data

STEP 1: Determine latest built date

Choose the latest year number in YearBuilt column and YearRemodAdd column as a new column, YearBuiltOrRe.

```
traindata$YearBuiltOrRe <- pmax(traindata$YearBuilt, traindata$YearRemo
dAdd)

testdata$YearBuiltOrRe <- pmax(testdata$YearBuilt, testdata$YearRemodAd
d)
```

STEP 2: Determine total bathrooms

Find out the total Bathrooms, use 0.5 for half bath, 1 for full bath.

```
traindata$TotalBath <- traindata$BsmtFullBath + (0.5 * traindata$BsmtHa
lfBath) +traindata$FullBath + (0.5 * traindata$HalfBath)

testdata$TotalBath <- testdata$BsmtFullBath + (0.5 * testdata$BsmtHalfB
ath) +testdata$FullBath + (0.5 * testdata$HalfBath)
```

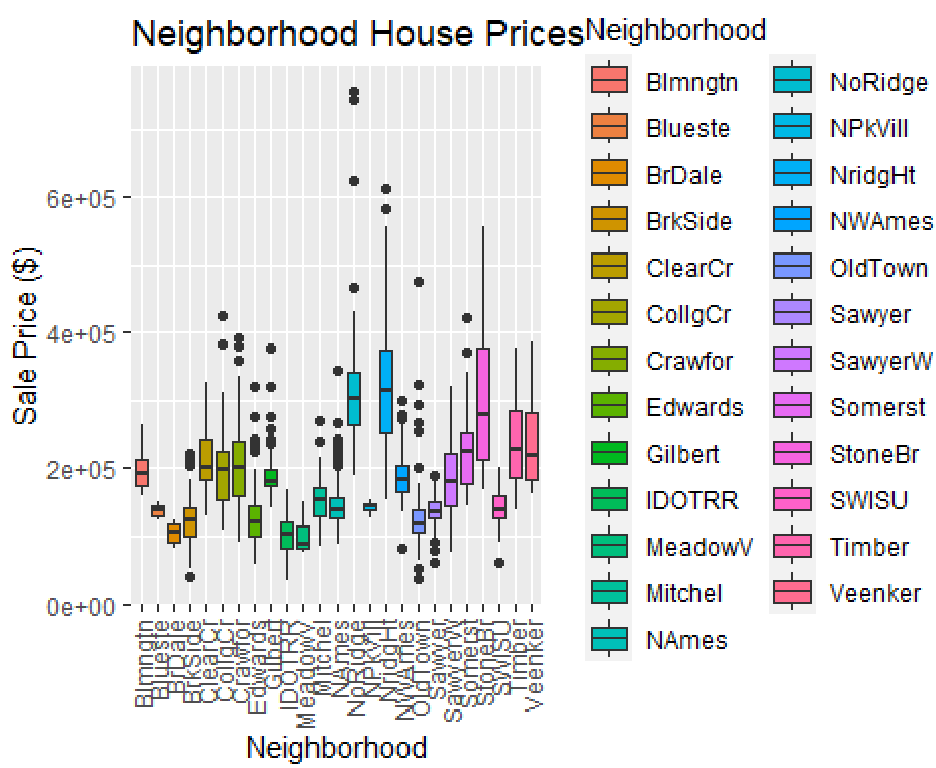
STEP 3: Rank neighborhoods with score

Find out the average sale price for house in each neighborhood. And replace neighborhood names with score from 1 to 10, rank by their average sale price.

```
unique_items_in_Neighborhood_train <- unique(traindata$Neighborhood)
unique_items_in_Neighborhood_train # Find out all Neighborhood in train
data

##  [1] "CollgCr" "Veenker" "Crawfor" "NoRidge" "Mitchel" "Somerst" "NW
Ames"
##  [8] "OldTown" "BrkSide" "Sawyer"  "NridgHt" "NAmes"   "SawyerW" "ID
OTRR"
## [15] "MeadowV" "Edwards" "Timber"  "Gilbert" "StoneBr" "ClearCr" "NP
kVill"
## [22] "Blmngtn" "BrDale"  "SWISU"   "Blueste"

# Find out prices and mean price in each neighborhood provided
neighborhood_prices <- traindata %>%
  group_by(Neighborhood) %>%
  summarise(mean_price = mean(SalePrice),
            prices = list(SalePrice))

print.data.frame(neighborhood_prices[, c("Neighborhood", "mean_price")])

##    Neighborhood mean_price
## 1       Blmngtn  194870.88
## 2       Blueste  137500.00
## 3        BrDale  104493.75
## 4       BrkSide  124834.05
## 5       ClearCr  212565.43
## 6       CollgCr  197965.77
## 7       Crawfor  210624.73
## 8       Edwards  128219.70
## 9       Gilbert  192854.51
## 10       IDOTRR  100123.78
## 11      MeadowV   98576.47
## 12      Mitchel  156270.12
```

```
## 13        NAmes  145847.08
## 14      NPkVill  142694.44
## 15       NWAmes  189050.07
## 16      NoRidge  335295.32
## 17      NridgHt  316270.62
## 18      OldTown  128225.30
## 19        SWISU  142591.36
## 20       Sawyer  136793.14
## 21      SawyerW  186555.80
## 22      Somerst  225379.84
## 23      StoneBr  310499.00
## 24       Timber  242247.45
## 25      Veenker  238772.73
```

```
# Graph Box plot to visually see how prices data locate in each neighbo
rhood
ggplot(traindata, aes(x = Neighborhood, y = SalePrice, fill = Neighborh
ood)) +
  geom_boxplot() +
  ggtitle("Neighborhood House Prices") +
  ylab("Sale Price ($)") +
  xlab("Neighborhood") +
  scale_fill_discrete(name = "Neighborhood") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5))
```




```
# Rank each neighborhood by SalePrice with score from 1 to 10
neighborhood_prices <- traindata %>%
```

```
  group_by(Neighborhood) %>%
  summarise(mean_price = mean(SalePrice)) %>%
  mutate(rank = rank(mean_price) / length(mean_price),
         NBscore = round(rank * 9) + 1)

print.data.frame(neighborhood_prices[, c("Neighborhood", "mean_price",
"NBscore")])

##    Neighborhood mean_price NBscore
## 1       Blmngtn  194870.88       7
## 2       Blueste  137500.00       4
## 3        BrDale  104493.75       2
## 4       BrkSide  124834.05       2
## 5       ClearCr  212565.43       8
## 6       CollgCr  197965.77       7
## 7       Crawfor  210624.73       7
## 8       Edwards  128219.70       3
## 9       Gilbert  192854.51       6
## 10       IDOTRR  100123.78       2
## 11      MeadowV   98576.47       1
## 12      Mitchel  156270.12       5
## 13        NAmes  145847.08       5
## 14      NPkVill  142694.44       5
## 15       NWAmes  189050.07       6
## 16      NoRidge  335295.32      10
## 17      NridgHt  316270.62      10
## 18      OldTown  128225.30       3
## 19        SWISU  142591.36       4
## 20       Sawyer  136793.14       4
## 21      SawyerW  186555.80       6
## 22      Somerst  225379.84       8
## 23      StoneBr  310499.00       9
## 24       Timber  242247.45       9
## 25      Veenker  238772.73       9

# Add Neighborhood Score in traindata
traindata <- traindata %>%
  left_join(neighborhood_prices[, c("Neighborhood", "NBscore")], by = "
Neighborhood")

unique_items_in_Neighborhood_test <- unique(testdata$Neighborhood)
unique_items_in_Neighborhood_test # Find out all Neighborhood in testda
ta

##  [1] "NAmes"   "Gilbert" "StoneBr" "BrDale"  "NPkVill" "NridgHt" "Bl
mngtn"
##  [8] "NoRidge" "Somerst" "SawyerW" "Sawyer"  "NWAmes"  "OldTown" "Br
kSide"
## [15] "ClearCr" "SWISU"   "Edwards" "CollgCr" "Crawfor" "Blueste" "ID
```

```
OTRR"
## [22] "Mitchel" "Timber"  "MeadowV" "Veenker"

lookup_table <- unique(traindata[, c("Neighborhood", "NBscore")])

# Merge the testdata with the lookup table to get the NBscore for each
neighborhood in testdata
testdata <- merge(testdata, lookup_table, by = "Neighborhood", all.x =
TRUE)
```

STEP 4: Check for the missing values

```
NA_values=data.frame(no_of_na_values=colSums(is.na(traindata)))
head(NA_values,26)

##                no_of_na_values
## Id                           0
## SalePrice                    0
## LotArea                      0
## Neighborhood                 0
## OverallQual                  0
## OverallCond                  0
## YearBuilt                    0
## YearRemodAdd                 0
## TotalBsmtSF                  0
## X1stFlrSF                    0
## X2ndFlrSF                    0
## GrLivArea                    0
## BsmtFullBath                 0
## BsmtHalfBath                 0
## FullBath                     0
## HalfBath                     0
## BedroomAbvGr                 0
## KitchenAbvGr                 0
## GarageCars                   0
## GarageArea                   0
## MoSold                       0
## YrSold                       0
## YearBuiltOrRe                0
## TotalBath                    0
## NBscore                      0
```

STEP 5: Clean all unnecessary columns

```
traindata <- traindata %>%
  select(-c(Neighborhood, BsmtFullBath, BsmtHalfBath, FullBath, HalfBat
h, MoSold, YearBuilt, YearRemodAdd))
```

```r
# Final look for traindata
head(traindata, 10)
```

```
##    Id SalePrice LotArea OverallQual OverallCond TotalBsmtSF X1stFlrS
F X2ndFlrSF
## 1   1    208500    8450           7           5         856        85
6       854
## 2   2    181500    9600           6           8        1262       126
2         0
## 3   3    223500   11250           7           5         920        92
0       866
## 4   4    140000    9550           7           5         756        96
1       756
## 5   5    250000   14260           8           5        1145       114
5      1053
## 6   6    143000   14115           5           5         796        79
6       566
## 7   7    307000   10084           8           5        1686       169
4         0
## 8   8    200000   10382           7           6        1107       110
7       983
## 9   9    129900    6120           7           5         952       102
2       752
## 10 10    118000    7420           5           6         991       107
7         0
##    GrLivArea BedroomAbvGr KitchenAbvGr GarageCars GarageArea YrSold
## 1       1710            3            1          2        548   2008
## 2       1262            3            1          2        460   2007
## 3       1786            3            1          2        608   2008
## 4       1717            3            1          3        642   2006
## 5       2198            4            1          3        836   2008
## 6       1362            1            1          2        480   2009
## 7       1694            3            1          2        636   2007
## 8       2090            3            1          2        484   2009
## 9       1774            2            2          2        468   2008
## 10      1077            2            2          1        205   2008
##    YearBuiltOrRe TotalBath NBscore
## 1           2003       3.5       7
## 2           1976       2.5       9
## 3           2002       3.5       7
## 4           1970       2.0       7
## 5           2000       3.5      10
## 6           1995       2.5       5
## 7           2005       3.0       8
## 8           1973       3.5       6
## 9           1950       2.0       3
## 10          1950       2.0       2
```

Divide Data to 2 Subset

Subset 1 is named in train_data with a ratio of 0.8 traindata, subset 2 is named in teast_data with a ratio of 0.2 traindata.

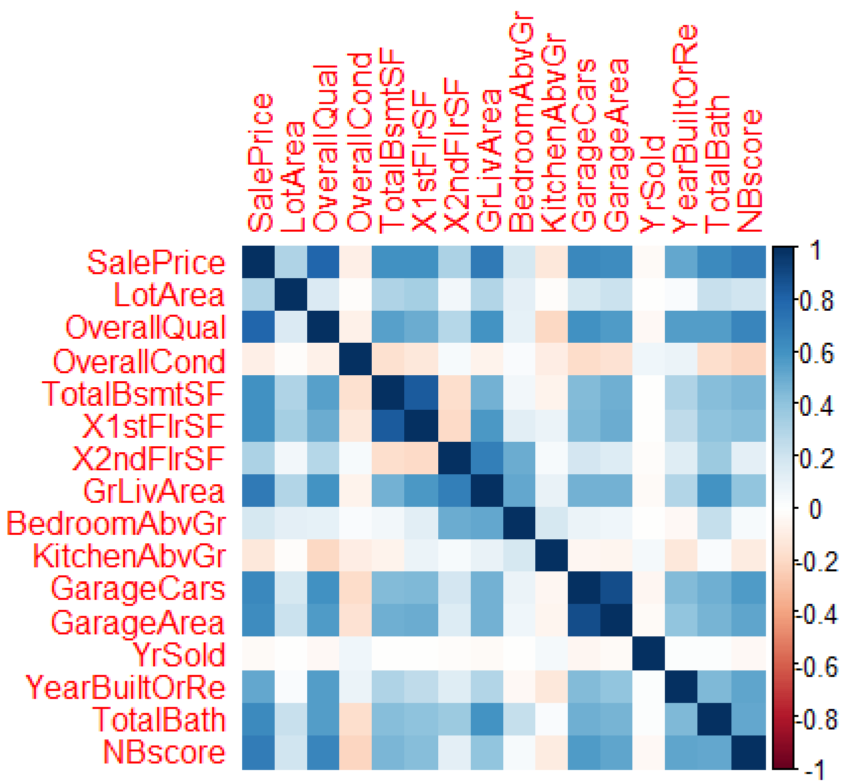
```
set.seed(700)   #  set seed to ensure you always have same random numbe
rs generated
sample = sample.split(traindata,SplitRatio = 0.8)
train_data =subset(traindata,sample ==TRUE)
test_data=subset(traindata, sample==FALSE)
```

Exploratory Data Analysis

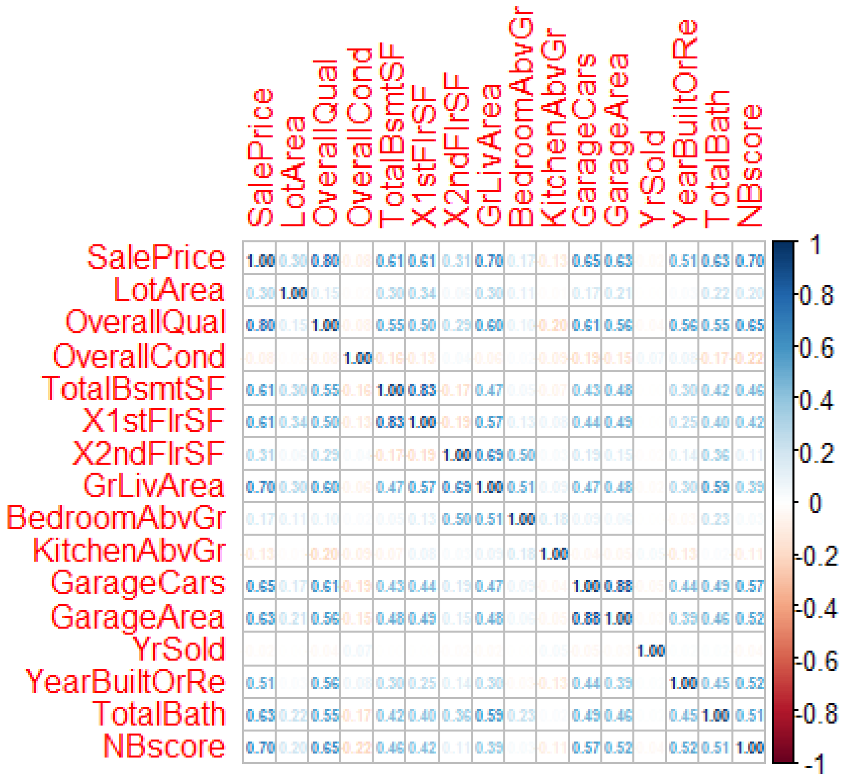
STEP 1: Correlation Plot

Determining the association between variables by their correlation.

```
cor_data=data.frame(train_data[,2:17])
correlation=cor(cor_data)
par(mfrow=c(1, 1))
corrplot(correlation,method="color")
```



```
corrplot(correlation,method="number", number.cex = 0.5)
```



According to our corrplot SalePrice is positively correlated with OverallQual, GrLivArea, GarageCars, YearBuiltOrRe, TotalBath, NBscore, LotArea, BedroomAbvGr, TotalBsmtSF, X1stFlrSF, X2ndFlrSF.

STEP 2: Scatter plots and Boxplots

Draw to Scatter plots and Boxplots to determine the relationship between these variables.

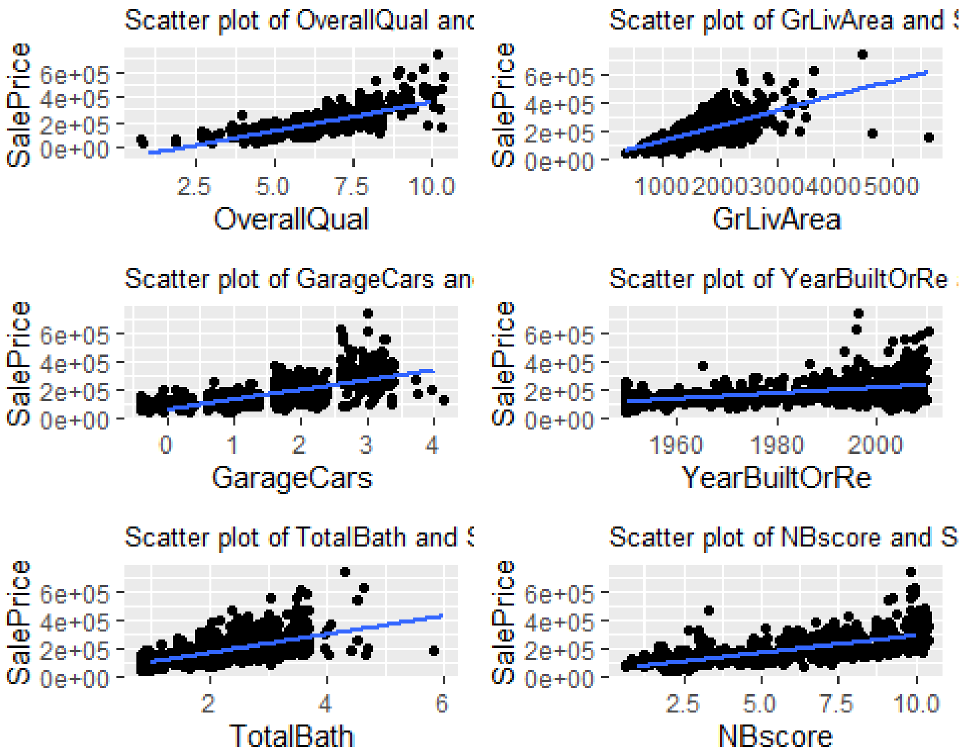
From following scatter plots, we conclude that the relationship between OverallQual, GrLivArea, GarageCars, YearBuiltOrRe, TotalBath, NBscore and LotArea is linear

```
p1=ggplot(data = train_data, aes(x = OverallQual, y = SalePrice)) +
  geom_jitter() +  geom_smooth(method = "lm", se = FALSE)+labs(title="S
catter plot of OverallQual and SalePrice", x="OverallQual",y="SalePrice
") + theme(plot.title = element_text(size = 10))
p2=ggplot(data = train_data, aes(x = GrLivArea, y = SalePrice)) +
  geom_jitter() +  geom_smooth(method = "lm", se = FALSE)+labs(title="S
catter plot of GrLivArea and SalePrice", x="GrLivArea",y="SalePrice") +
```

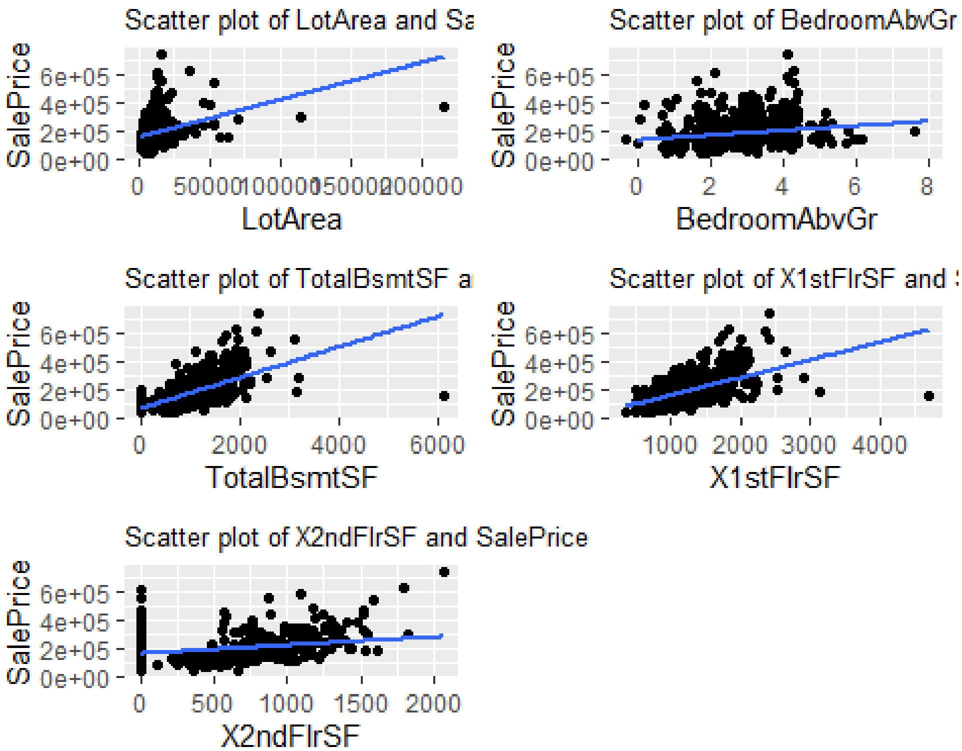
```
 theme(plot.title = element_text(size = 10))
p3=ggplot(data = train_data, aes(x = GarageCars, y = SalePrice)) +
  geom_jitter() +  geom_smooth(method = "lm", se = FALSE)+labs(title="S
catter plot of GarageCars and SalePrice", x="GarageCars",y="SalePrice")
 + theme(plot.title = element_text(size = 10))
p4=ggplot(data = train_data, aes(x = YearBuiltOrRe, y = SalePrice)) +
  geom_jitter() +  geom_smooth(method = "lm", se = FALSE)+labs(title="S
catter plot of YearBuiltOrRe and SalePrice", x="YearBuiltOrRe",y="SaleP
rice") + theme(plot.title = element_text(size = 10))
p5=ggplot(data = train_data, aes(x = TotalBath, y = SalePrice)) +
  geom_jitter() +  geom_smooth(method = "lm", se = FALSE)+labs(title="S
catter plot of TotalBath and SalePrice", x="TotalBath",y="SalePrice") +
 theme(plot.title = element_text(size = 10))
p6=ggplot(data = train_data, aes(x = NBscore, y = SalePrice)) +
  geom_jitter() +  geom_smooth(method = "lm", se = FALSE)+labs(title="S
catter plot of NBscore and SalePrice", x="NBscore",y="SalePrice") + the
me(plot.title = element_text(size = 10))
p7=ggplot(data = train_data, aes(x = LotArea, y = SalePrice)) +
  geom_jitter() +  geom_smooth(method = "lm", se = FALSE)+labs(title="S
catter plot of LotArea and SalePrice", x="LotArea",y="SalePrice") + the
me(plot.title = element_text(size = 10))
p8=ggplot(data = train_data, aes(x = BedroomAbvGr, y = SalePrice)) +
  geom_jitter() +  geom_smooth(method = "lm", se = FALSE)+labs(title="S
catter plot of BedroomAbvGr and SalePrice", x="BedroomAbvGr",y="SalePri
ce") + theme(plot.title = element_text(size = 10))
p9=ggplot(data = train_data, aes(x = TotalBsmtSF, y = SalePrice)) +
  geom_jitter() +  geom_smooth(method = "lm", se = FALSE)+labs(title="S
catter plot of TotalBsmtSF and SalePrice", x="TotalBsmtSF",y="SalePrice
") + theme(plot.title = element_text(size = 10))
p10=ggplot(data = train_data, aes(x = X1stFlrSF, y = SalePrice)) +
  geom_jitter() +  geom_smooth(method = "lm", se = FALSE)+labs(title="S
catter plot of X1stFlrSF and SalePrice", x="X1stFlrSF",y="SalePrice") +
 theme(plot.title = element_text(size = 10))
p11=ggplot(data = train_data, aes(x = X2ndFlrSF, y = SalePrice)) +
  geom_jitter() +  geom_smooth(method = "lm", se = FALSE)+labs(title="S
catter plot of X2ndFlrSF and SalePrice", x="X2ndFlrSF",y="SalePrice") +
 theme(plot.title = element_text(size = 10))
```

Scatter Plots

```
grid.arrange(p1,p2,p3,p4,p5,p6,nrow=3)

## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



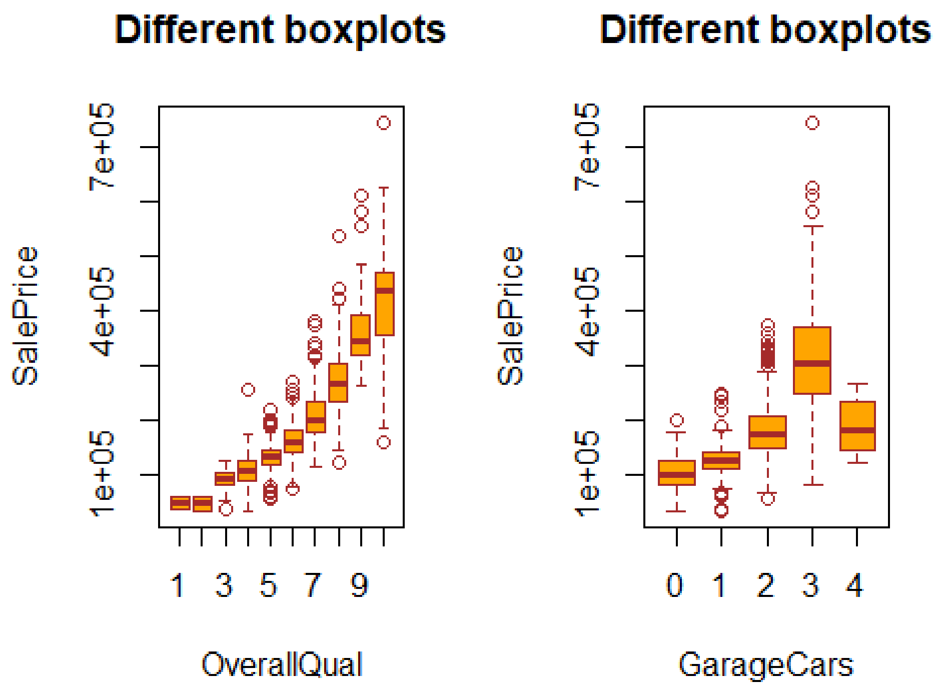

```
grid.arrange(p7,p8,p9,p10,p11,nrow=3)

## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



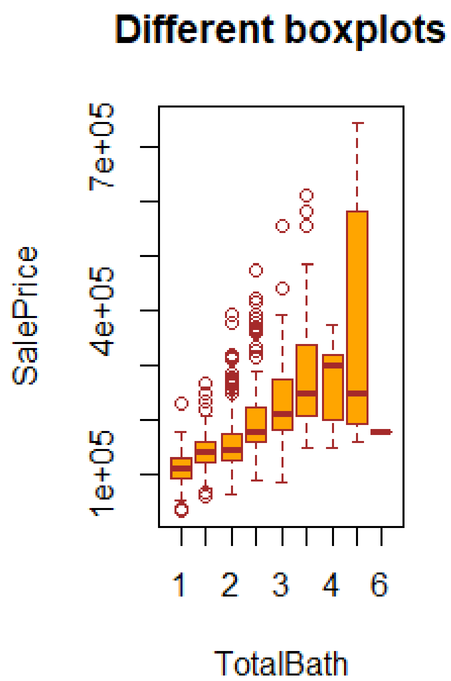
Box Plots

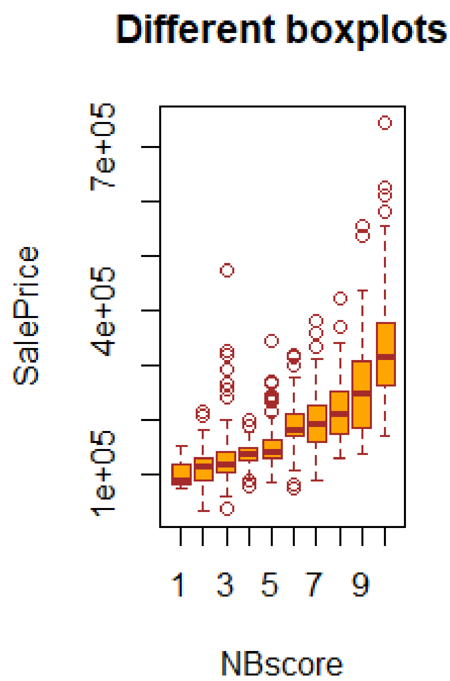
For the 4 categorical variables(OverallQual, GarageCars, TotalBath, and NBscore) we draw boxplots to understand the relationship.

```
par(mfrow=c(1, 2))
boxplot(SalePrice~OverallQual,data=train_data,main="Different boxplots",
 xlab="OverallQual",ylab="SalePrice",col="orange",border="brown")
boxplot(SalePrice~GarageCars,data=train_data,main="Different boxplots",
 xlab="GarageCars",ylab="SalePrice",col="orange",border="brown")
```




```
boxplot(SalePrice~TotalBath,data=train_data,main="Different boxplots",
xlab="TotalBath",ylab="SalePrice",col="orange",border="brown")
boxplot(SalePrice~NBscore,data=train_data,main="Different boxplots", xl
ab="NBscore",ylab="SalePrice",col="orange",border="brown")
```

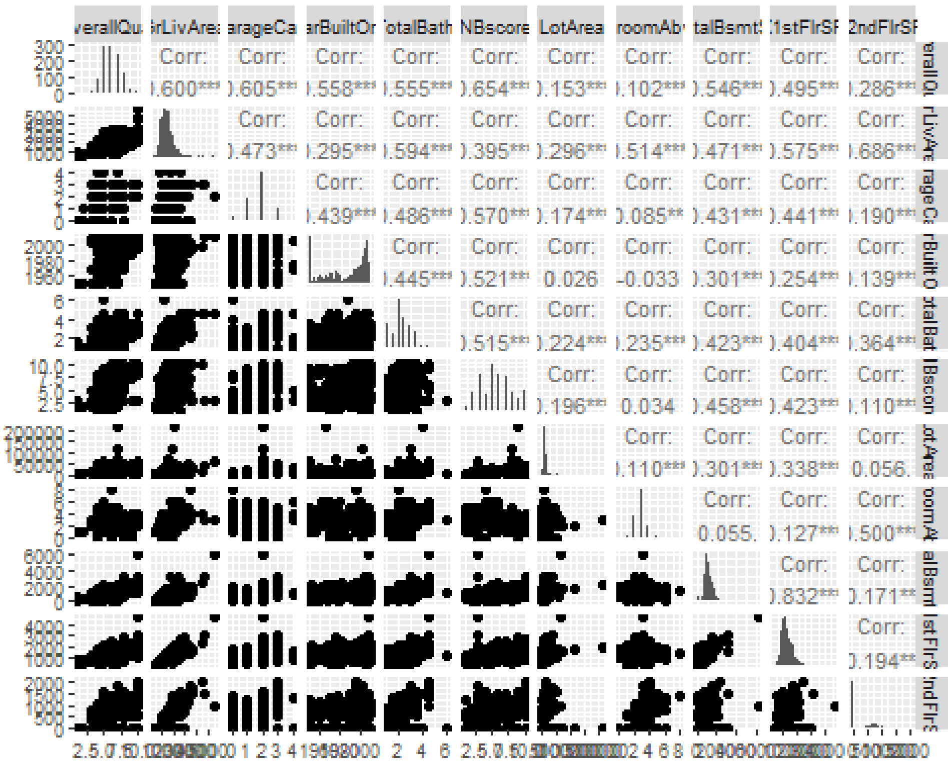




There is a relationship between price and categorical variables, OverallQual, GarageCars, TotalBath, and NBscore.

STEP 3: Performing ggpair plot.

```
ggpairs(train_data,
        columns= c("OverallQual","GrLivArea","GarageCars","YearBuiltOrRe","TotalBath","NBscore","LotArea","BedroomAbvGr","TotalBsmtSF","X1stFlrSF", "X2ndFlrSF"),
        diag = list(continuous = wrap("barDiag", cex = 0.5)),
        upper = list(continuous = wrap("cor", size = 3))) +
  theme_grey(base_size = 8)

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



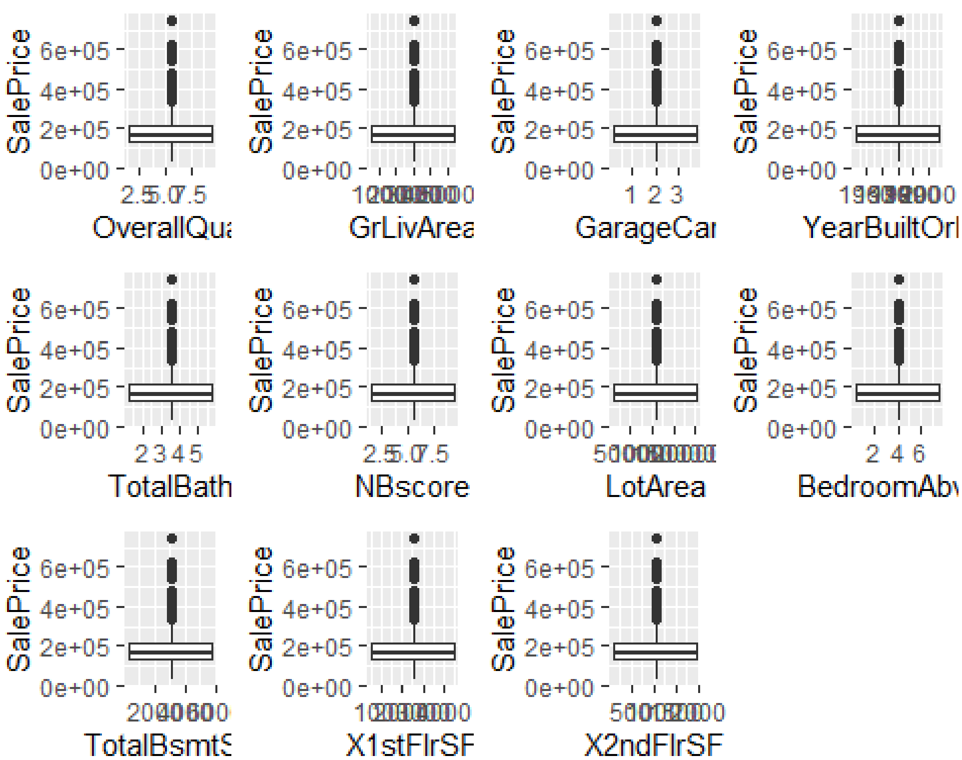
STEP 4: Determine outliers with boxplots

Check and analysis for outliers in the dependent variable(price) using a boxplot.

a. Identify Outliers by drawing boxplot

```
b1 <- ggplot(data=train_data)+geom_boxplot(aes(x=OverallQual,y=SalePric
e))
b2 <- ggplot(data=train_data)+geom_boxplot(aes(x=GrLivArea,y=SalePrice))
b3 <- ggplot(data=train_data)+geom_boxplot(aes(x=GarageCars,y=SalePric
e))
b4 <- ggplot(data=train_data)+geom_boxplot(aes(x=YearBuiltOrRe,y=SalePr
ice))
b5 <- ggplot(data=train_data)+geom_boxplot(aes(x=TotalBath,y=SalePrice))
b6 <- ggplot(data=train_data)+geom_boxplot(aes(x=NBscore,y=SalePrice))
b7 <- ggplot(data=train_data)+geom_boxplot(aes(x=LotArea,y=SalePrice))
b8 <- ggplot(data=train_data)+geom_boxplot(aes(x=BedroomAbvGr,y=SalePri
ce))
b9 <- ggplot(data=train_data)+geom_boxplot(aes(x=TotalBsmtSF,y=SalePric
e))
b10 <- ggplot(data=train_data)+geom_boxplot(aes(x=X1stFlrSF,y=SalePric
e))
b11 <- ggplot(data=train_data)+geom_boxplot(aes(x=X2ndFlrSF,y=SalePric
```

```
e))

grid.arrange(b1, b2, b3, b4, b5, b6, b7, b8, b9, b10, b11, nrow=3)

## Warning: Continuous x aesthetic
## i did you forget `aes(group = ...)`?
## Continuous x aesthetic
## i did you forget `aes(group = ...)`?
## Continuous x aesthetic
## i did you forget `aes(group = ...)`?
## Continuous x aesthetic
## i did you forget `aes(group = ...)`?
## Continuous x aesthetic
## i did you forget `aes(group = ...)`?
## Continuous x aesthetic
## i did you forget `aes(group = ...)`?
## Continuous x aesthetic
## i did you forget `aes(group = ...)`?
## Continuous x aesthetic
## i did you forget `aes(group = ...)`?
## Continuous x aesthetic
## i did you forget `aes(group = ...)`?
## Continuous x aesthetic
## i did you forget `aes(group = ...)`?
## Continuous x aesthetic
## i did you forget `aes(group = ...)`?
```

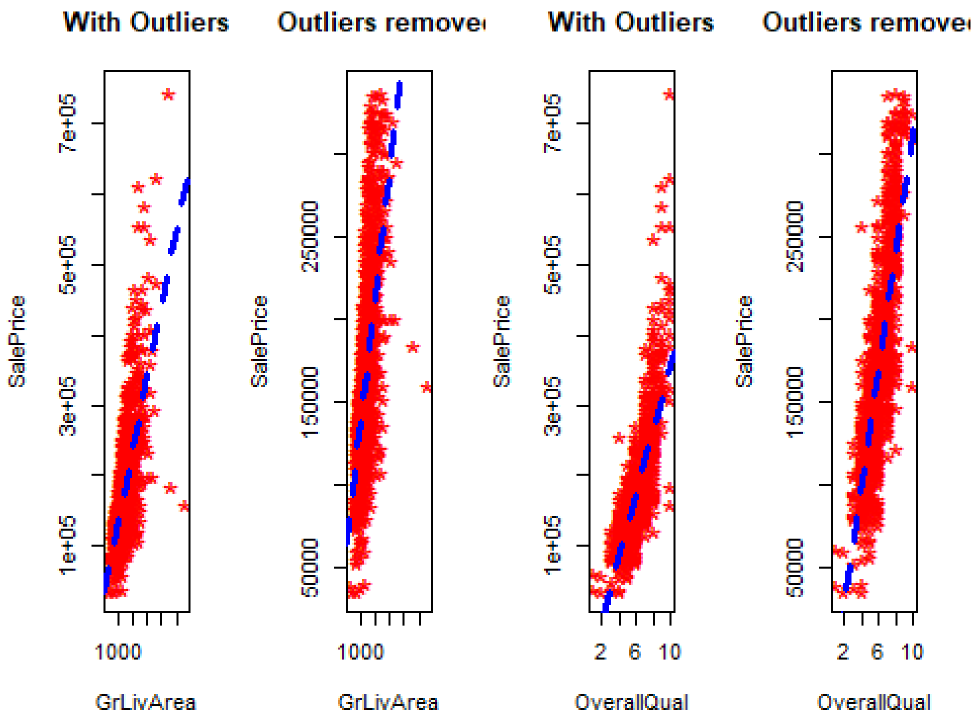


b. Create new data set without all outliers named in train_data1

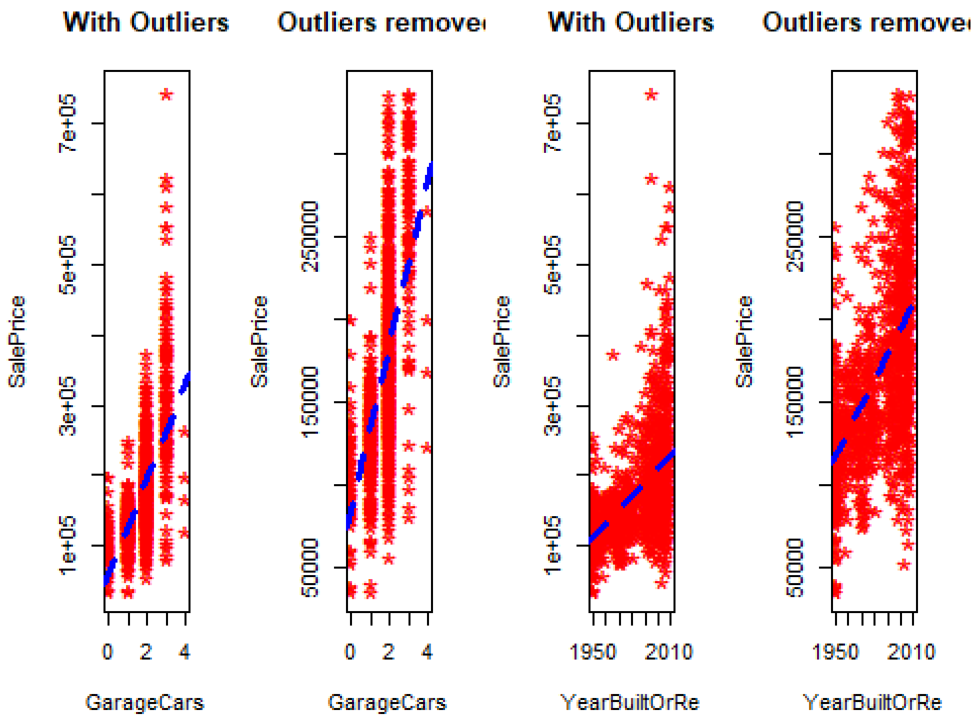
```
outliers=boxplot(train_data$SalePrice,plot=FALSE)$out
outliers_data=train_data[which(train_data$SalePrice %in% outliers),]
train_data1= train_data[-which(train_data$SalePrice %in% outliers),]
length(outliers)

## [1] 51
```

c. Analysis datas with Outliers and without Outliers in scatter plot

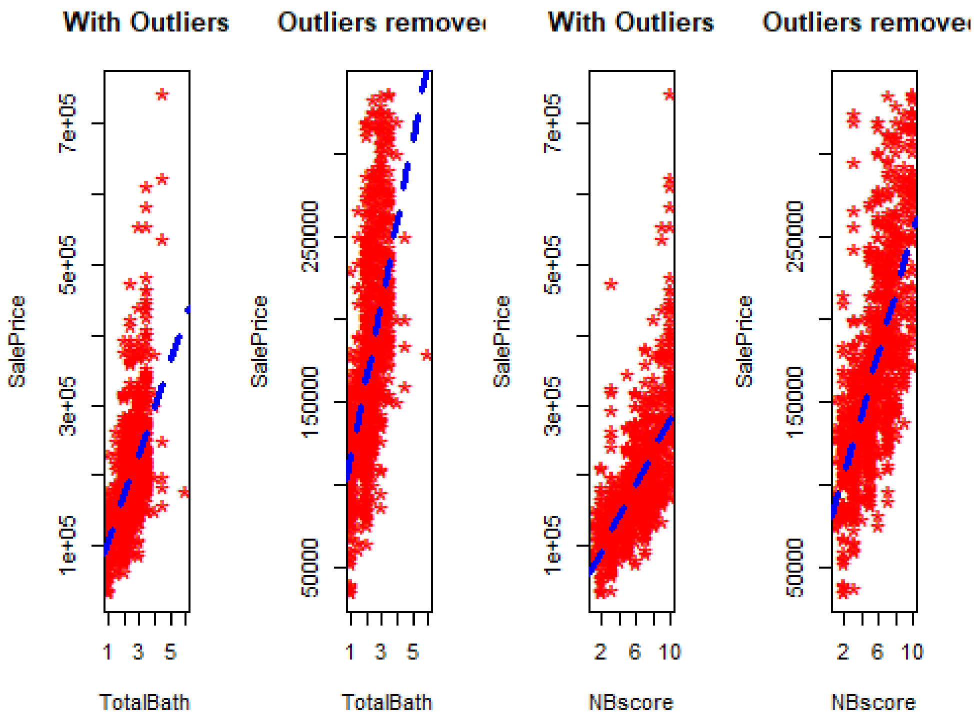
```
par(mfrow=c(1, 4))

# OverallQual
plot(train_data$GrLivArea, train_data$SalePrice, main="With Outliers",
xlab="GrLivArea", ylab="SalePrice", pch="*", col="red", cex=2)
abline(lm(SalePrice ~ GrLivArea, data=train_data), col="blue", lwd=3, l
ty=2)
plot(train_data1$GrLivArea, train_data1$SalePrice, main="Outliers remov
ed", xlab="GrLivArea", ylab="SalePrice", pch="*", col="red", cex=2)
abline(lm(SalePrice ~GrLivArea, data=train_data1), col="blue", lwd=3, l
ty=2)

# GrLivArea
plot(train_data$OverallQual, train_data$SalePrice, main="With Outliers",
 xlab="OverallQual", ylab="SalePrice", pch="*", col="red", cex=2)
abline(lm(SalePrice ~ OverallQual, data=train_data), col="blue", lwd=3,
 lty=2)
plot(train_data1$OverallQual, train_data1$SalePrice, main="Outliers rem
oved", xlab="OverallQual", ylab="SalePrice", pch="*", col="red", cex=2)
abline(lm(SalePrice ~OverallQual, data=train_data1), col="blue", lwd=3,
 lty=2)
```



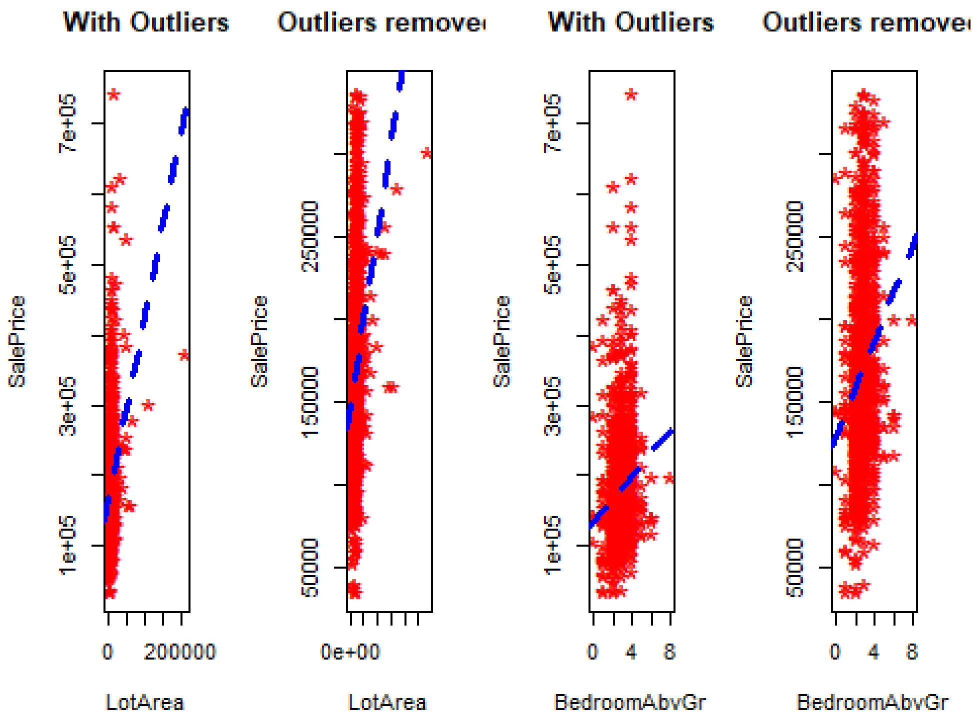

```
# GarageCars
plot(train_data$GarageCars, train_data$SalePrice, main="With Outliers",
 xlab="GarageCars", ylab="SalePrice", pch="*", col="red", cex=2)
abline(lm(SalePrice ~ GarageCars, data=train_data), col="blue", lwd=3,
lty=2)
plot(train_data1$GarageCars, train_data1$SalePrice, main="Outliers remo
ved", xlab="GarageCars", ylab="SalePrice", pch="*", col="red", cex=2)
abline(lm(SalePrice ~GarageCars, data=train_data1), col="blue", lwd=3,
lty=2)

# YearBuiltOrRe
plot(train_data$YearBuiltOrRe, train_data$SalePrice, main="With Outlier
s", xlab="YearBuiltOrRe", ylab="SalePrice", pch="*", col="red", cex=2)
abline(lm(SalePrice ~ YearBuiltOrRe, data=train_data), col="blue", lwd=
3, lty=2)
plot(train_data1$YearBuiltOrRe, train_data1$SalePrice, main="Outliers r
emoved", xlab="YearBuiltOrRe", ylab="SalePrice", pch="*", col="red", ce
x=2)
abline(lm(SalePrice ~ YearBuiltOrRe, data=train_data1), col="blue", lwd
=3, lty=2)
```



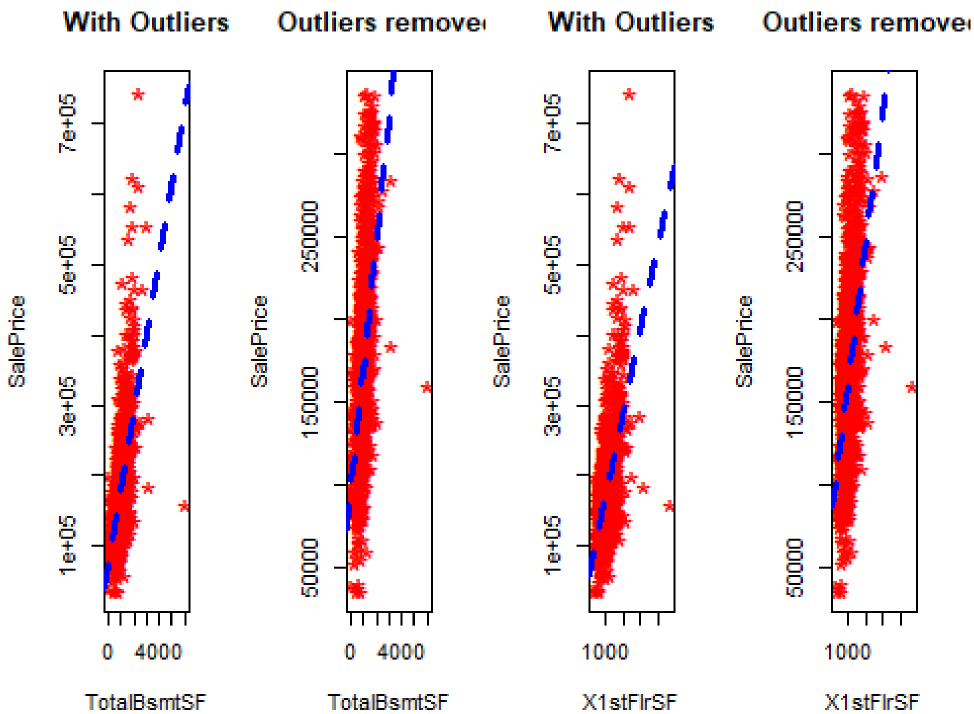

```
# TotalBath
plot(train_data$TotalBath, train_data$SalePrice, main="With Outliers",
xlab="TotalBath", ylab="SalePrice", pch="*", col="red", cex=2)
abline(lm(SalePrice ~ TotalBath, data=train_data), col="blue", lwd=3, l
ty=2)
plot(train_data1$TotalBath, train_data1$SalePrice, main="Outliers remov
ed", xlab="TotalBath", ylab="SalePrice", pch="*", col="red", cex=2)
abline(lm(SalePrice ~ TotalBath, data=train_data1), col="blue", lwd=3,
lty=2)

# NBscore
plot(train_data$NBscore, train_data$SalePrice, main="With Outliers", xl
ab="NBscore", ylab="SalePrice", pch="*", col="red", cex=2)
abline(lm(SalePrice ~ NBscore, data=train_data), col="blue", lwd=3, lty
=2)
plot(train_data1$NBscore, train_data1$SalePrice, main="Outliers removed
", xlab="NBscore", ylab="SalePrice", pch="*", col="red", cex=2)
abline(lm(SalePrice ~ NBscore, data=train_data1), col="blue", lwd=3, lt
y=2)
```



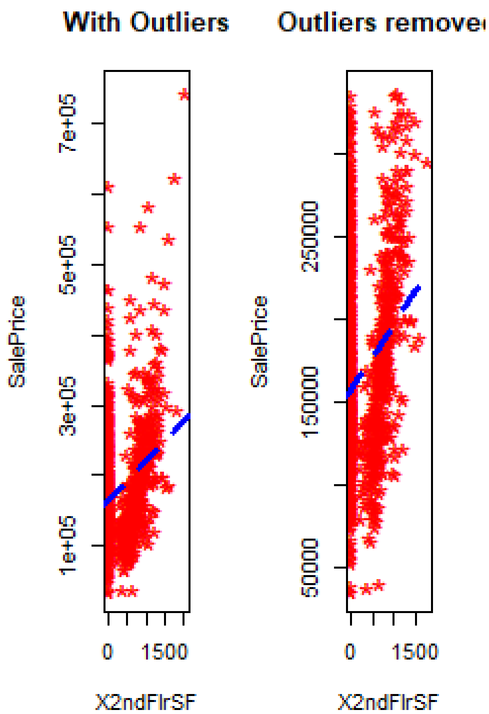

```
# LotArea
plot(train_data$LotArea, train_data$SalePrice, main="With Outliers", xl
ab="LotArea", ylab="SalePrice", pch="*", col="red", cex=2)
abline(lm(SalePrice ~ LotArea, data=train_data), col="blue", lwd=3, lty
=2)
plot(train_data1$LotArea, train_data1$SalePrice, main="Outliers removed
", xlab="LotArea", ylab="SalePrice", pch="*", col="red", cex=2)
abline(lm(SalePrice ~ LotArea, data=train_data1), col="blue", lwd=3, lt
y=2)

# BedroomAbvGr
plot(train_data$BedroomAbvGr, train_data$SalePrice, main="With Outliers
", xlab="BedroomAbvGr", ylab="SalePrice", pch="*", col="red", cex=2)
abline(lm(SalePrice ~ BedroomAbvGr, data=train_data), col="blue", lwd=3,
 lty=2)
plot(train_data1$BedroomAbvGr, train_data1$SalePrice, main="Outliers re
moved", xlab="BedroomAbvGr", ylab="SalePrice", pch="*", col="red", cex=
2)
abline(lm(SalePrice ~ BedroomAbvGr, data=train_data1), col="blue", lwd=
3, lty=2)
```




```
# TotalBsmtSF
plot(train_data$TotalBsmtSF, train_data$SalePrice, main="With Outliers",
 xlab="TotalBsmtSF", ylab="SalePrice", pch="*", col="red", cex=2)
abline(lm(SalePrice ~ TotalBsmtSF, data=train_data), col="blue", lwd=3,
 lty=2)
plot(train_data1$TotalBsmtSF, train_data1$SalePrice, main="Outliers rem
oved", xlab="TotalBsmtSF", ylab="SalePrice", pch="*", col="red", cex=2)
abline(lm(SalePrice ~ TotalBsmtSF, data=train_data1), col="blue", lwd=3,
 lty=2)

# X1stFlrSF
plot(train_data$X1stFlrSF, train_data$SalePrice, main="With Outliers",
xlab="X1stFlrSF", ylab="SalePrice", pch="*", col="red", cex=2)
abline(lm(SalePrice ~ X1stFlrSF, data=train_data), col="blue", lwd=3, l
ty=2)
plot(train_data1$X1stFlrSF, train_data1$SalePrice, main="Outliers remov
ed", xlab="X1stFlrSF", ylab="SalePrice", pch="*", col="red", cex=2)
abline(lm(SalePrice ~ X1stFlrSF, data=train_data1), col="blue", lwd=3,
lty=2)
```




```
# X2ndFlrSF
plot(train_data$X2ndFlrSF, train_data$SalePrice, main="With Outliers",
xlab="X2ndFlrSF", ylab="SalePrice", pch="*", col="red", cex=2)
abline(lm(SalePrice ~ X2ndFlrSF, data=train_data), col="blue", lwd=3, l
ty=2)
plot(train_data1$X2ndFlrSF, train_data1$SalePrice, main="Outliers remov
ed", xlab="X2ndFlrSF", ylab="SalePrice", pch="*", col="red", cex=2)
abline(lm(SalePrice ~ X2ndFlrSF, data=train_data1), col="blue", lwd=3,
lty=2)
```



Modeling

STEP 1: Model on the all train data

SalePrice, OverallQual, GrLivArea, GarageCars, YearBuiltOrRe, TotalBath, NBscore, LotArea, BedroomAbvGr, TotalBsmtSF, X1stFlrSF, X2ndFlrSF were considered for the full model based on the corrplot.

Model 1: linear fit of all variables.

```
model1=lm(data=train_data,SalePrice~OverallQual+GrLivArea+GarageCars+Ye
arBuiltOrRe+TotalBath+NBscore+LotArea+BedroomAbvGr+TotalBsmtSF+X1stFlrS
F+X2ndFlrSF)
summary(model1)

##
## Call:
## lm(formula = SalePrice ~ OverallQual + GrLivArea + GarageCars +
##     YearBuiltOrRe + TotalBath + NBscore + LotArea + BedroomAbvGr +
##     TotalBsmtSF + X1stFlrSF + X2ndFlrSF, data = train_data)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -399547  -18030   -1831   15041  278310
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -3.603e+05  1.303e+05  -2.766 0.005766 **
## OverallQual      1.684e+04  1.329e+03  12.675  < 2e-16 ***
## GrLivArea        2.948e+01  2.346e+01   1.256 0.209226
## GarageCars       1.099e+04  1.946e+03   5.645 2.10e-08 ***
## YearBuiltOrRe    1.416e+02  6.696e+01   2.114 0.034702 *
## TotalBath        6.846e+03  1.918e+03   3.570 0.000372 ***
## NBscore          7.347e+03  6.723e+02  10.927  < 2e-16 ***
## LotArea          5.442e-01  1.290e-01   4.219 2.66e-05 ***
## BedroomAbvGr    -5.360e+03  1.630e+03  -3.288 0.001041 **
## TotalBsmtSF      1.366e+01  4.603e+00   2.967 0.003074 **
## X1stFlrSF        2.340e+01  2.409e+01   0.972 0.331449
## X2ndFlrSF        1.482e+01  2.388e+01   0.620 0.535155
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35850 on 1104 degrees of freedom
## Multiple R-squared:  0.8007, Adjusted R-squared:  0.7987
## F-statistic: 403.1 on 11 and 1104 DF,  p-value: < 2.2e-16
```

From the relationship between these variables appear to be strong as shown by Adjusted R-Suared value, **0.7932** and the probability. Also conclude from the p-value that GrLivArea, X1stFlrSF, X2ndFlrSF are not a significant variable for the prediction of price. By drop this variables a lower Adjusted R-Suared value, **0.7662** appeared. Thus we shouldn't drop these varibles in our regression model.

Model 2: linear fit of part variables.

```
model2=lm(data=train_data,SalePrice~OverallQual+GarageCars+YearBuiltOrR
e+TotalBath+NBscore+LotArea+BedroomAbvGr+TotalBsmtSF)
summary(model2)

##
## Call:
## lm(formula = SalePrice ~ OverallQual + GarageCars + YearBuiltOrRe +
##     TotalBath + NBscore + LotArea + BedroomAbvGr + TotalBsmtSF,
##     data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -309814  -19738   -3462   15290  348969
##
## Coefficients:
```

```
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -3.007e+05  1.401e+05  -2.146   0.0321 *
## OverallQual    2.295e+04  1.327e+03  17.300  < 2e-16 ***
## GarageCars     1.463e+04  2.065e+03   7.087 2.44e-12 ***
## YearBuiltOrRe  9.807e+01  7.202e+01   1.362   0.1735
## TotalBath      1.403e+04  1.961e+03   7.157 1.50e-12 ***
## NBscore        6.398e+03  7.186e+02   8.904  < 2e-16 ***
## LotArea        8.498e-01  1.365e-01   6.224 6.87e-10 ***
## BedroomAbvGr   6.120e+03  1.471e+03   4.161 3.42e-05 ***
## TotalBsmtSF    2.615e+01  3.256e+00   8.033 2.42e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 38630 on 1107 degrees of freedom
## Multiple R-squared:  0.7678, Adjusted R-squared:  0.7662
## F-statistic: 457.7 on 8 and 1107 DF,  p-value: < 2.2e-16
```

By drop this variables a lower Adjusted R-Suared value, **0.7662** appeared. Thus we shouldn't drop these varibles in our regression model.

STEP 2: Model outliers from variables

Model the entire training data and decide on the retention of outliers in different variables.

Model 1: with all outliers

```
# with all outliers
model1=lm(data=train_data,SalePrice~OverallQual+GrLivArea+GarageCars+YearBuiltOrRe+TotalBath+NBscore+LotArea+BedroomAbvGr+TotalBsmtSF+X1stFlrSF+X2ndFlrSF)
summary(model1)
```

```
##
## Call:
## lm(formula = SalePrice ~ OverallQual + GrLivArea + GarageCars +
##     YearBuiltOrRe + TotalBath + NBscore + LotArea + BedroomAbvGr +
##     TotalBsmtSF + X1stFlrSF + X2ndFlrSF, data = train_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -399547  -18030   -1831   15041  278310
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -3.603e+05  1.303e+05  -2.766 0.005766 **
## OverallQual    1.684e+04  1.329e+03  12.675  < 2e-16 ***
```

```
## GrLivArea      2.948e+01  2.346e+01   1.256 0.209226
## GarageCars     1.099e+04  1.946e+03   5.645 2.10e-08 ***
## YearBuiltOrRe  1.416e+02  6.696e+01   2.114 0.034702 *
## TotalBath      6.846e+03  1.918e+03   3.570 0.000372 ***
## NBscore        7.347e+03  6.723e+02  10.927  < 2e-16 ***
## LotArea        5.442e-01  1.290e-01   4.219 2.66e-05 ***
## BedroomAbvGr  -5.360e+03  1.630e+03  -3.288 0.001041 **
## TotalBsmtSF    1.366e+01  4.603e+00   2.967 0.003074 **
## X1stFlrSF      2.340e+01  2.409e+01   0.972 0.331449
## X2ndFlrSF      1.482e+01  2.388e+01   0.620 0.535155
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 35850 on 1104 degrees of freedom
## Multiple R-squared:  0.8007, Adjusted R-squared:  0.7987
## F-statistic: 403.1 on 11 and 1104 DF,  p-value: < 2.2e-16
```

Model 3: without all outliers

```
# without all outliers


model3=lm(data=train_data1,SalePrice~OverallQual+GrLivArea+GarageCars+Y
earBuiltOrRe+TotalBath+NBscore+LotArea+BedroomAbvGr+TotalBsmtSF+X1stFlr
SF+X2ndFlrSF)
summary(model3)
```

```
##
## Call:
## lm(formula = SalePrice ~ OverallQual + GrLivArea + GarageCars +
##     YearBuiltOrRe + TotalBath + NBscore + LotArea + BedroomAbvGr +
##     TotalBsmtSF + X1stFlrSF + X2ndFlrSF, data = train_data1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -278891  -13628    -516   13462   99663
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -5.080e+05  9.284e+04  -5.472 5.57e-08 ***
## OverallQual    1.406e+04  9.721e+02  14.465  < 2e-16 ***
## GrLivArea     -9.263e+00  1.784e+01  -0.519 0.603796
## GarageCars     9.390e+03  1.392e+03   6.747 2.49e-11 ***
## YearBuiltOrRe  2.343e+02  4.776e+01   4.907 1.07e-06 ***
## TotalBath      8.075e+03  1.378e+03   5.860 6.19e-09 ***
## NBscore        6.528e+03  4.850e+02  13.460  < 2e-16 ***
## LotArea        5.125e-01  1.368e-01   3.747 0.000188 ***
## BedroomAbvGr   3.744e+02  1.195e+03   0.313 0.754051
## TotalBsmtSF    9.175e+00  3.333e+00   2.753 0.006014 **
```

```
## X1stFlrSF      4.054e+01  1.824e+01   2.222 0.026488 *
## X2ndFlrSF      3.485e+01  1.813e+01   1.922 0.054832 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25400 on 1053 degrees of freedom
## Multiple R-squared:  0.8137, Adjusted R-squared:  0.8118
## F-statistic: 418.1 on 11 and 1053 DF,  p-value: < 2.2e-16
```

By comparing linear model 1 with all outliers and model 3 without all outliers. The summary from model 1 and model 3 showed that we should clean all outliers. And by take a deep look at P value in T test for all variables, we should not clean the outliers in BedroomAbvGr, TotalBsmtSF, X2ndFlrSF and GrLivArea .

```
outliers <- unlist(lapply(train_data[, c("OverallQual", "GarageCars", "
YearBuiltOrRe", "TotalBath", "NBscore", "LotArea", "X2ndFlrSF","X1stFlr
SF")], function(x) boxplot(x, plot=FALSE)$out))


train_data2 <- train_data
for (col in c("OverallQual", "GarageCars", "YearBuiltOrRe", "TotalBath",
 "NBscore", "LotArea", "X2ndFlrSF","X1stFlrSF")) {
  train_data2 <- train_data2[!(train_data2[, col] %in% outliers), ]
}
```

Model 4: with parts outliers

```
# with parts outliers
model4=lm(data=train_data2,SalePrice~OverallQual+GarageCars+YearBuiltOr
Re+TotalBath+NBscore+LotArea+BedroomAbvGr+TotalBsmtSF+X1stFlrSF+X2ndFlr
SF)
summary(model4)

##
## Call:
## lm(formula = SalePrice ~ OverallQual + GarageCars + YearBuiltOrRe +
##     TotalBath + NBscore + LotArea + BedroomAbvGr + TotalBsmtSF +
##     X1stFlrSF + X2ndFlrSF, data = train_data2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -135039  -17970   -1293   18360  101222
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -7.887e+05  2.602e+05  -3.031 0.002644 **
## OverallQual     1.494e+04  2.392e+03   6.247 1.39e-09 ***
```

```
## GarageCars     1.445e+04  3.382e+03   4.271 2.59e-05 ***
## YearBuiltOrRe  3.426e+02  1.337e+02   2.563 0.010863 *
## TotalBath      1.374e+04  4.012e+03   3.424 0.000702 ***
## NBscore        4.500e+03  1.066e+03   4.223 3.19e-05 ***
## LotArea        4.010e+00  5.757e-01   6.965 2.00e-11 ***
## BedroomAbvGr  -1.094e+04  2.666e+03  -4.104 5.21e-05 ***
## TotalBsmtSF    1.042e+01  9.082e+00   1.147 0.252173
## X1stFlrSF      6.686e+01  9.787e+00   6.832 4.50e-11 ***
## X2ndFlrSF      4.014e+01  6.479e+00   6.195 1.87e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28920 on 307 degrees of freedom
## Multiple R-squared:  0.8375, Adjusted R-squared:  0.8322
## F-statistic: 158.2 on 10 and 307 DF,  p-value: < 2.2e-16
```

As concluded from the Adjusted R-squared value of 0.8227, the relationship between these variables appear to be quite strong.

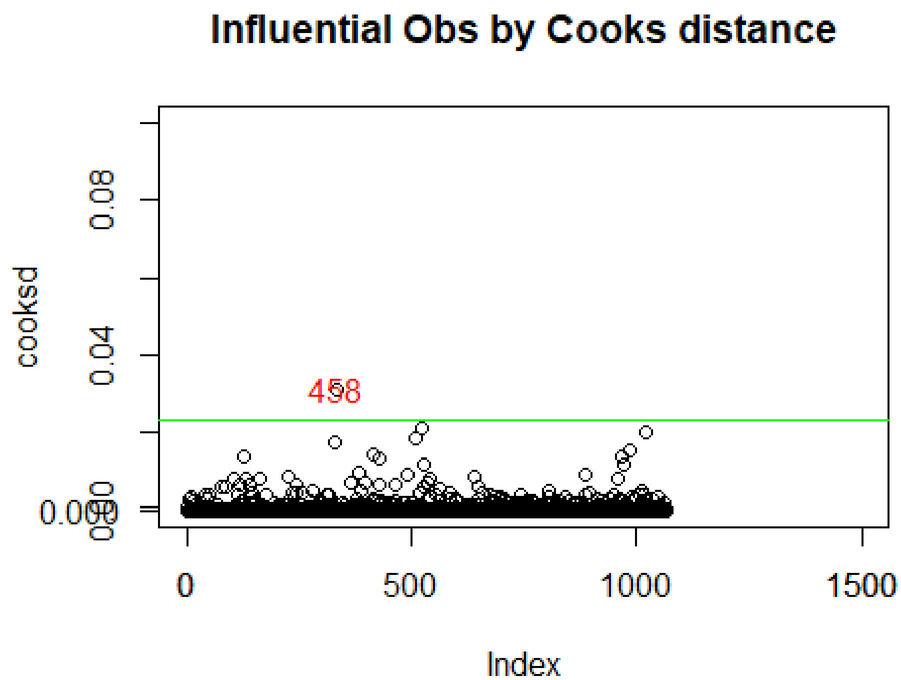
STEP 3: Detect Influential Points

If we label an observation as an outlier based on only one feature (even if it's not that important), it could lead us to draw incorrect conclusions. Instead, it's better to consider all the different features (or X's) when we're trying to determine whether a particular entity (like a row or observation) is an extreme value. The Cook's distance is a useful tool that can help us do this and identify which features are most relevant.

```
cooksd <- cooks.distance(model3)
mean(cooksd)

## [1] 0.005874981
```

Plot the cook's distance.

```
par(mfrow=c(1, 1))
plot(cooksd, main="Influential Obs by Cooks distance",xlim=c(0,1500),yl
im=c(0,0.1))
axis(1, at=seq(0, 1500, 1500))
axis(2, at=seq(0, 0.001, 0.001), las=1)
abline(h = 4*mean(cooksd, na.rm=T), col="green")
text(x=1:length(cooksd)+1,y=cooksd,labels=ifelse(cooksd>4*mean(cooksd,n
a.rm=T),names(cooksd),""), col="red")
```



Find the influential points in the data.

```
influential <- as.numeric(names(cooksd)[(cooksd > 4*mean(cooksd, na.rm=
T))])  # influential row numbers
head(train_data2[influential, ])

##        Id SalePrice LotArea OverallQual OverallCond TotalBsmtSF X1stFl
rSF
## NA    NA         NA       NA           NA           NA           NA
 NA
## NA.1 NA         NA       NA           NA           NA           NA
 NA
## NA.2 NA         NA       NA           NA           NA           NA
 NA
##        X2ndFlrSF GrLivArea BedroomAbvGr KitchenAbvGr GarageCars Garage
Area YrSold
## NA           NA         NA           NA           NA          NA
  NA      NA
## NA.1         NA         NA           NA           NA          NA
  NA      NA
## NA.2         NA         NA           NA           NA          NA
  NA      NA
##        YearBuiltOrRe TotalBath NBscore
## NA                NA         NA       NA
```

```
## NA.1              NA         NA       NA
## NA.2              NA         NA       NA
```

```
influential_data=train_data2[influential, ]
```

Take out the influential outliers.

```
influencial_outliers=inner_join(outliers_data,influential_data)

## Joining with `by = join_by(Id, SalePrice, LotArea, OverallQual, Over
allCond,
## TotalBsmtSF, X1stFlrSF, X2ndFlrSF, GrLivArea, BedroomAbvGr, KitchenA
bvGr,
## GarageCars, GarageArea, YrSold, YearBuiltOrRe, TotalBath, NBscore)`

influencial_outliers

##  [1] Id            SalePrice     LotArea       OverallQual   Overall
Cond
##  [6] TotalBsmtSF   X1stFlrSF     X2ndFlrSF     GrLivArea     Bedroom
AbvGr
## [11] KitchenAbvGr  GarageCars    GarageArea    YrSold        YearBui
ltOrRe
## [16] TotalBath     NBscore
## <0 行> (或0-长度的row.names)
```

We have **17 observations** which are outliers yet influential hence we need to keep these outliers.

Modify the Influential Outliers

Modify the data excluding the outliers and including only the influential outliers.

```
train_data3=rbind(train_data2,influencial_outliers)
```

STEP 4: Model with Influential Outliers

Modelling using the train data which includes influential_outliers

Model 5: with influential_outliers

```
# Model 5: with influential_outliers
model5=lm(data=train_data3,SalePrice~OverallQual+GarageCars+YearBuiltOr
Re+TotalBath+NBscore+LotArea+BedroomAbvGr+TotalBsmtSF+X1stFlrSF+X2ndFlr
```

```
SF)
summary(model5)
```

```
##
## Call:
## lm(formula = SalePrice ~ OverallQual + GarageCars + YearBuiltOrRe +
##     TotalBath + NBscore + LotArea + BedroomAbvGr + TotalBsmtSF +
##     X1stFlrSF + X2ndFlrSF, data = train_data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -135039  -17970   -1293   18360  101222
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -7.887e+05  2.602e+05  -3.031 0.002644 **
## OverallQual    1.494e+04  2.392e+03   6.247 1.39e-09 ***
## GarageCars     1.445e+04  3.382e+03   4.271 2.59e-05 ***
## YearBuiltOrRe  3.426e+02  1.337e+02   2.563 0.010863 *
## TotalBath      1.374e+04  4.012e+03   3.424 0.000702 ***
## NBscore        4.500e+03  1.066e+03   4.223 3.19e-05 ***
## LotArea        4.010e+00  5.757e-01   6.965 2.00e-11 ***
## BedroomAbvGr  -1.094e+04  2.666e+03  -4.104 5.21e-05 ***
## TotalBsmtSF    1.042e+01  9.082e+00   1.147 0.252173
## X1stFlrSF      6.686e+01  9.787e+00   6.832 4.50e-11 ***
## X2ndFlrSF      4.014e+01  6.479e+00   6.195 1.87e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 28920 on 307 degrees of freedom
## Multiple R-squared:  0.8375, Adjusted R-squared:  0.8322
## F-statistic: 158.2 on 10 and 307 DF,  p-value: < 2.2e-16
```

Model 6: model 5 without 2 strars or less variables

```
model6=lm(data=train_data3,SalePrice~OverallQual+GarageCars+TotalBath+N
Bscore+LotArea+BedroomAbvGr+X1stFlrSF+X2ndFlrSF)
summary(model6)
```

```
##
## Call:
## lm(formula = SalePrice ~ OverallQual + GarageCars + TotalBath +
##     NBscore + LotArea + BedroomAbvGr + X1stFlrSF + X2ndFlrSF,
##     data = train_data3)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -133210  -18199    -815   17525  103127
##
```

```
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.264e+05  1.269e+04  -9.956  < 2e-16 ***
## OverallQual   1.821e+04  2.085e+03   8.734  < 2e-16 ***
## GarageCars    1.562e+04  3.385e+03   4.614 5.80e-06 ***
## TotalBath     1.414e+04  4.043e+03   3.498 0.000538 ***
## NBscore       5.036e+03  1.056e+03   4.770 2.85e-06 ***
## LotArea       3.936e+00  5.780e-01   6.811 5.08e-11 ***
## BedroomAbvGr -1.040e+04  2.648e+03  -3.926 0.000107 ***
## X1stFlrSF     7.074e+01  7.708e+00   9.177  < 2e-16 ***
## X2ndFlrSF     3.572e+01  6.039e+00   5.915 8.79e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 29180 on 309 degrees of freedom
## Multiple R-squared:  0.8335, Adjusted R-squared:  0.8292
## F-statistic: 193.3 on 8 and 309 DF,  p-value: < 2.2e-16
```

The relationship between above variables appear to be very strong as shown by R-Suared value and the probability. Even I try fitting the model including a few other variables which we left out, the R squared value won't increase. As a conclude from the p-value that all variables are relevantly significant with two to three stars for the prediction of price. Hence we keep all variable in **model 5**.

As concluded from the Adjusted R-squared value from model 5 with 0.8322, the relationship between these variables appear to be vary strong.

Accuracy of Model

```
pred=model5$fitted.values


tally_table=data.frame(actual=train_data3$SalePrice, predicted=pred)

mape=mean(abs(tally_table$actual-tally_table$predicted)/tally_table$act
ual)
accuracy=1-mape
accuracy

## [1] 0.8895409
```

We see that the accuracy of train_data3 (0.8 of the overall cleaned traindata) is 88.95%

```
pred_test=predict(newdata=test_data,model5)
```

```
tally_table_1=data.frame(actual=test_data$SalePrice, predicted=pred_tes
t)

mape_test=mean(abs(tally_table_1$actual-tally_table_1$predicted)/tally_
table_1$actual)
accuracy_test=1-mape_test
accuracy_test

## [1] 0.8223009
```

We see that the accuracy of test_data (0.2 of the overall traindata) is 82.23%. Thus our model can predict price with an accuracy of 82.23%