Assignment 1

Alicia Chimeno & Silvia Ferrer

2023-11-18

## Clear space and Libraries

##Load dataset

#setwd("C:/Users/Silvia/OneDrive - Universitat Politècnica de Catalunya/Escritorio/UPC/MASTER DS/1A/SIM/SIM\_assignment\_1")  
setwd("/Users/ali/Desktop/MASTER/SIM/PROJECT 1")  
train<-read.delim("train.csv", sep=',')

## Explore dataset and Preproces dataset

dim(train)

## [1] 1460 81

#names(train)  
str(train)

## '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" ...  
## $ Electrical : chr "SBrkr" "SBrkr" "SBrkr" "SBrkr" ...  
## $ X1stFlrSF : int 856 1262 920 961 1145 796 1694 1107 1022 1077 ...  
## $ X2ndFlrSF : int 854 0 866 756 1053 566 0 983 752 0 ...  
## $ LowQualFinSF : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ GrLivArea : int 1710 1262 1786 1717 2198 1362 1694 2090 1774 1077 ...  
## $ 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 ...  
## $ KitchenQual : chr "Gd" "TA" "Gd" "Gd" ...  
## $ TotRmsAbvGrd : int 8 6 6 7 9 5 7 7 8 5 ...  
## $ Functional : chr "Typ" "Typ" "Typ" "Typ" ...  
## $ Fireplaces : int 0 1 1 1 1 0 1 2 2 2 ...  
## $ FireplaceQu : chr NA "TA" "TA" "Gd" ...  
## $ GarageType : chr "Attchd" "Attchd" "Attchd" "Detchd" ...  
## $ GarageYrBlt : int 2003 1976 2001 1998 2000 1993 2004 1973 1931 1939 ...  
## $ GarageFinish : chr "RFn" "RFn" "RFn" "Unf" ...  
## $ GarageCars : int 2 2 2 3 3 2 2 2 2 1 ...  
## $ GarageArea : int 548 460 608 642 836 480 636 484 468 205 ...  
## $ GarageQual : chr "TA" "TA" "TA" "TA" ...  
## $ GarageCond : chr "TA" "TA" "TA" "TA" ...  
## $ PavedDrive : chr "Y" "Y" "Y" "Y" ...  
## $ WoodDeckSF : int 0 298 0 0 192 40 255 235 90 0 ...  
## $ OpenPorchSF : int 61 0 42 35 84 30 57 204 0 4 ...  
## $ EnclosedPorch: int 0 0 0 272 0 0 0 228 205 0 ...  
## $ X3SsnPorch : int 0 0 0 0 0 320 0 0 0 0 ...  
## $ ScreenPorch : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ PoolArea : int 0 0 0 0 0 0 0 0 0 0 ...  
## $ PoolQC : chr NA NA NA NA ...  
## $ Fence : chr NA NA NA NA ...  
## $ MiscFeature : chr NA NA NA NA ...  
## $ MiscVal : int 0 0 0 0 0 700 0 350 0 0 ...  
## $ MoSold : int 2 5 9 2 12 10 8 11 4 1 ...  
## $ YrSold : int 2008 2007 2008 2006 2008 2009 2007 2009 2008 2008 ...  
## $ SaleType : chr "WD" "WD" "WD" "WD" ...  
## $ SaleCondition: chr "Normal" "Normal" "Normal" "Abnorml" ...  
## $ SalePrice : int 208500 181500 223500 140000 250000 143000 307000 200000 129900 118000 ...

#skim(train)

## Record incorrect NA’s

train$Alley[which(is.na(train$Alley))] <- "No alley access"  
train$BsmtQual[which(is.na(train$BsmtQual))] <- "No Basement"  
train$BsmtCond[which(is.na(train$BsmtCond))] <- "No Basement"  
train$BsmtExposure[which(is.na(train$BsmtExposure))] <- "No Basement"  
train$BsmtFinType1[which(is.na(train$BsmtFinType1))] <- "No Basement"  
train$BsmtFinType2[which(is.na(train$BsmtFinType2))] <- "No Basement"  
train$FireplaceQu[which(is.na(train$FireplaceQu))] <- "No Fireplace"  
train$GarageType[which(is.na(train$GarageType))] <- "No Garage"  
train$GarageFinish[which(is.na(train$GarageFinish))] <- "No Garage"  
train$GarageCond[which(is.na(train$GarageCond))] <- "No Garage"  
train$PoolQC[which(is.na(train$PoolQC))] <- "No Pool"  
train$Fence[which(is.na(train$Fence))] <- "No Fence"  
train$MiscFeature[which(is.na(train$MiscFeature))] <- "None"

# Recode variables to factor

# Recode: character to factor  
char\_cols <- which(sapply(train, is.character))  
train[, char\_cols] <- lapply(train[, char\_cols], as.factor)   
  
# Recode: numeric to factor  
train$MSSubClass <- factor(train$MSSubClass)  
train$MoSold <- factor(train$MoSold) # month

## Target variable exploration “SalePrice”

We asses a normality test to the target variable. We reject null hypothesis (H0: variable is normal), therfore we shall not consider that the target variable is normally distributed.

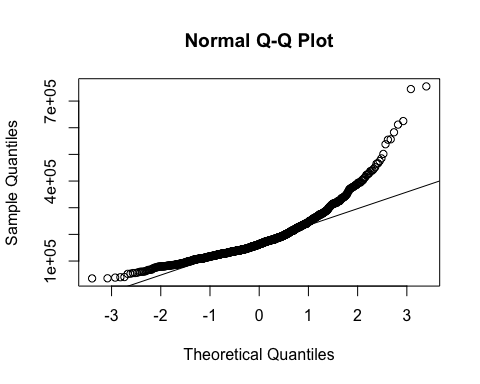
res.con <- condes(train,81)  
head(res.con$quali)

## R2 p.value  
## Neighborhood 0.5455750 1.558600e-225  
## ExterQual 0.4773878 1.439551e-204  
## BsmtQual 0.4649938 8.158548e-196  
## KitchenQual 0.4565986 3.032213e-192  
## GarageFinish 0.3058737 6.228747e-115  
## FireplaceQu 0.2939608 2.971217e-107

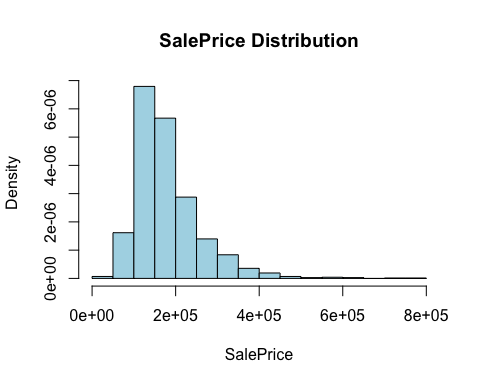
shapiro.test(train$SalePrice)

##   
## Shapiro-Wilk normality test  
##   
## data: train$SalePrice  
## W = 0.86967, p-value < 2.2e-16

qqnorm(train$SalePrice)  
qqline(train$SalePrice)



hist(train$SalePrice, prob = TRUE, col = 'lightblue', main = 'SalePrice Distribution', xlab = 'SalePrice')



#lines(density(train$SalePrice), col = 'red', lwd = 2)

## Selection of 10 factor variables and numerical variables

We select the first 10th categorical variables that are more associated with the target variable.

vect<-row.names(res.con$quali)  
vect[1:10]

## [1] "Neighborhood" "ExterQual" "BsmtQual" "KitchenQual" "GarageFinish"  
## [6] "FireplaceQu" "Foundation" "GarageType" "MSSubClass" "BsmtFinType1"

factor\_train<-train[,c(vect[1:10])]  
num\_cols <- names(train)[sapply(train, is.numeric)]  
train2 <- train[,c(num\_cols, vect[1:10])]  
dim(train2)

## [1] 1460 46

## Variable analysis

# Analysis functions

To proceede with the predictor analysis, we shall create the following functions:

## numeric variable description function  
numeric\_description <- function(variable, n\_breaks) {  
 cat("Summary:\n")  
 print(summary(variable))  
   
 cat("\nCount of missing values:",sum(is.na(variable)),"\n")  
 column\_name <- sub(".+\\$", "", deparse(substitute(variable)))  
 hist(variable, breaks = n\_breaks, freq = F, main = paste("Histograma de", column\_name), xlab = column\_name)  
 curve(dnorm(x, mean(variable), sd(variable)), add = T)  
   
 # Normality test with Shapiro-Wilk  
 print(shapiro.test(variable))  
}  
  
## analyze outliers functions  
analyze\_outliers <- function(data, column\_name) {  
 sm <- summary(data[[column\_name]])  
 iqr <- sm["3rd Qu."] - sm["1st Qu."]  
   
 # Mild Outliers  
 mild\_ub <- sm["3rd Qu."] + 1.5 \* iqr  
 mild\_lb <- sm["1st Qu."] - 1.5 \* iqr  
   
 mild\_outliers <- length(which(data[[column\_name]] > mild\_ub | data[[column\_name]] < mild\_lb))  
 cat("Number of mild outliers:", mild\_outliers, "\n")  
   
 # Plotting mild outliers  
 Boxplot(data[[column\_name]], main = paste("Outlier Analysis for", column\_name),  
 ylab = column\_name, outline = TRUE)  
 abline(h = mild\_ub, col = "orange", lwd = 2)  
 abline(h = mild\_lb, col = "orange", lwd = 2)  
   
 # Severe Outliers  
 severe\_ub <- sm["3rd Qu."] + 3 \* iqr  
 severe\_lb <- sm["1st Qu."] - 3 \* iqr  
 severe\_outliers <- length(which(data[[column\_name]] > severe\_ub | data[[column\_name]] < severe\_lb))  
 cat("Number of severe outliers:", severe\_outliers, "\n")  
 # Plotting severe outliers  
 abline(h = severe\_ub, col = "red", lwd = 2.5)  
 abline(h = severe\_lb, col = "red", lwd = 2.5)  
}  
# count outliers (new column)  
  
train2$univ\_outl\_count <- 0 # initialize  
  
# count outliers  
update\_outliers\_count <- function(dataframe, column\_name) {  
 df <- dataframe  
 sm <- summary(df[[column\_name]])  
 iqr <- sm["3rd Qu."] - sm["1st Qu."]  
 # Severe Outliers  
 severe\_ub <- sm["3rd Qu."] + 3 \* iqr  
 severe\_lb <- sm["1st Qu."] - 3 \* iqr  
 severe\_outliers\_id <- which(df[[column\_name]] > severe\_ub | df[[column\_name]] < severe\_lb)  
 df$univ\_outl\_count[severe\_outliers\_id] <- df$univ\_outl\_count[severe\_outliers\_id] + 1  
 return(df)  
}

# EDA and univariate outliers of each variable

skim(train2) # overall exploration

Data summary

|  |  |
| --- | --- |
| Name | train2 |
| Number of rows | 1460 |
| Number of columns | 47 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Column type frequency: |  |
| factor | 10 |
| numeric | 37 |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |  |
| Group variables | None |

**Variable type: factor**

| skim\_variable | n\_missing | complete\_rate | ordered | n\_unique | top\_counts |
| --- | --- | --- | --- | --- | --- |
| Neighborhood | 0 | 1 | FALSE | 25 | NAm: 225, Col: 150, Old: 113, Edw: 100 |
| ExterQual | 0 | 1 | FALSE | 4 | TA: 906, Gd: 488, Ex: 52, Fa: 14 |
| BsmtQual | 0 | 1 | FALSE | 5 | TA: 649, Gd: 618, Ex: 121, No : 37 |
| KitchenQual | 0 | 1 | FALSE | 4 | TA: 735, Gd: 586, Ex: 100, Fa: 39 |
| GarageFinish | 0 | 1 | FALSE | 4 | Unf: 605, RFn: 422, Fin: 352, No : 81 |
| FireplaceQu | 0 | 1 | FALSE | 6 | No : 690, Gd: 380, TA: 313, Fa: 33 |
| Foundation | 0 | 1 | FALSE | 6 | PCo: 647, CBl: 634, Brk: 146, Sla: 24 |
| GarageType | 0 | 1 | FALSE | 7 | Att: 870, Det: 387, Bui: 88, No : 81 |
| MSSubClass | 0 | 1 | FALSE | 15 | 20: 536, 60: 299, 50: 144, 120: 87 |
| BsmtFinType1 | 0 | 1 | FALSE | 7 | Unf: 430, GLQ: 418, ALQ: 220, BLQ: 148 |

**Variable type: numeric**

| skim\_variable | n\_missing | complete\_rate | mean | sd | p0 | p25 | p50 | p75 | p100 | hist |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Id | 0 | 1.00 | 730.50 | 421.61 | 1 | 365.75 | 730.5 | 1095.25 | 1460 | ▇▇▇▇▇ |
| LotFrontage | 259 | 0.82 | 70.05 | 24.28 | 21 | 59.00 | 69.0 | 80.00 | 313 | ▇▃▁▁▁ |
| LotArea | 0 | 1.00 | 10516.83 | 9981.26 | 1300 | 7553.50 | 9478.5 | 11601.50 | 215245 | ▇▁▁▁▁ |
| OverallQual | 0 | 1.00 | 6.10 | 1.38 | 1 | 5.00 | 6.0 | 7.00 | 10 | ▁▂▇▅▁ |
| OverallCond | 0 | 1.00 | 5.58 | 1.11 | 1 | 5.00 | 5.0 | 6.00 | 9 | ▁▁▇▅▁ |
| YearBuilt | 0 | 1.00 | 1971.27 | 30.20 | 1872 | 1954.00 | 1973.0 | 2000.00 | 2010 | ▁▂▃▆▇ |
| YearRemodAdd | 0 | 1.00 | 1984.87 | 20.65 | 1950 | 1967.00 | 1994.0 | 2004.00 | 2010 | ▅▂▂▃▇ |
| MasVnrArea | 8 | 0.99 | 103.69 | 181.07 | 0 | 0.00 | 0.0 | 166.00 | 1600 | ▇▁▁▁▁ |
| BsmtFinSF1 | 0 | 1.00 | 443.64 | 456.10 | 0 | 0.00 | 383.5 | 712.25 | 5644 | ▇▁▁▁▁ |
| BsmtFinSF2 | 0 | 1.00 | 46.55 | 161.32 | 0 | 0.00 | 0.0 | 0.00 | 1474 | ▇▁▁▁▁ |
| BsmtUnfSF | 0 | 1.00 | 567.24 | 441.87 | 0 | 223.00 | 477.5 | 808.00 | 2336 | ▇▅▂▁▁ |
| TotalBsmtSF | 0 | 1.00 | 1057.43 | 438.71 | 0 | 795.75 | 991.5 | 1298.25 | 6110 | ▇▃▁▁▁ |
| X1stFlrSF | 0 | 1.00 | 1162.63 | 386.59 | 334 | 882.00 | 1087.0 | 1391.25 | 4692 | ▇▅▁▁▁ |
| X2ndFlrSF | 0 | 1.00 | 346.99 | 436.53 | 0 | 0.00 | 0.0 | 728.00 | 2065 | ▇▃▂▁▁ |
| LowQualFinSF | 0 | 1.00 | 5.84 | 48.62 | 0 | 0.00 | 0.0 | 0.00 | 572 | ▇▁▁▁▁ |
| GrLivArea | 0 | 1.00 | 1515.46 | 525.48 | 334 | 1129.50 | 1464.0 | 1776.75 | 5642 | ▇▇▁▁▁ |
| BsmtFullBath | 0 | 1.00 | 0.43 | 0.52 | 0 | 0.00 | 0.0 | 1.00 | 3 | ▇▆▁▁▁ |
| BsmtHalfBath | 0 | 1.00 | 0.06 | 0.24 | 0 | 0.00 | 0.0 | 0.00 | 2 | ▇▁▁▁▁ |
| FullBath | 0 | 1.00 | 1.57 | 0.55 | 0 | 1.00 | 2.0 | 2.00 | 3 | ▁▇▁▇▁ |
| HalfBath | 0 | 1.00 | 0.38 | 0.50 | 0 | 0.00 | 0.0 | 1.00 | 2 | ▇▁▅▁▁ |
| BedroomAbvGr | 0 | 1.00 | 2.87 | 0.82 | 0 | 2.00 | 3.0 | 3.00 | 8 | ▁▇▂▁▁ |
| KitchenAbvGr | 0 | 1.00 | 1.05 | 0.22 | 0 | 1.00 | 1.0 | 1.00 | 3 | ▁▇▁▁▁ |
| TotRmsAbvGrd | 0 | 1.00 | 6.52 | 1.63 | 2 | 5.00 | 6.0 | 7.00 | 14 | ▂▇▇▁▁ |
| Fireplaces | 0 | 1.00 | 0.61 | 0.64 | 0 | 0.00 | 1.0 | 1.00 | 3 | ▇▇▁▁▁ |
| GarageYrBlt | 81 | 0.94 | 1978.51 | 24.69 | 1900 | 1961.00 | 1980.0 | 2002.00 | 2010 | ▁▁▅▅▇ |
| GarageCars | 0 | 1.00 | 1.77 | 0.75 | 0 | 1.00 | 2.0 | 2.00 | 4 | ▁▃▇▂▁ |
| GarageArea | 0 | 1.00 | 472.98 | 213.80 | 0 | 334.50 | 480.0 | 576.00 | 1418 | ▂▇▃▁▁ |
| WoodDeckSF | 0 | 1.00 | 94.24 | 125.34 | 0 | 0.00 | 0.0 | 168.00 | 857 | ▇▂▁▁▁ |
| OpenPorchSF | 0 | 1.00 | 46.66 | 66.26 | 0 | 0.00 | 25.0 | 68.00 | 547 | ▇▁▁▁▁ |
| EnclosedPorch | 0 | 1.00 | 21.95 | 61.12 | 0 | 0.00 | 0.0 | 0.00 | 552 | ▇▁▁▁▁ |
| X3SsnPorch | 0 | 1.00 | 3.41 | 29.32 | 0 | 0.00 | 0.0 | 0.00 | 508 | ▇▁▁▁▁ |
| ScreenPorch | 0 | 1.00 | 15.06 | 55.76 | 0 | 0.00 | 0.0 | 0.00 | 480 | ▇▁▁▁▁ |
| PoolArea | 0 | 1.00 | 2.76 | 40.18 | 0 | 0.00 | 0.0 | 0.00 | 738 | ▇▁▁▁▁ |
| MiscVal | 0 | 1.00 | 43.49 | 496.12 | 0 | 0.00 | 0.0 | 0.00 | 15500 | ▇▁▁▁▁ |
| YrSold | 0 | 1.00 | 2007.82 | 1.33 | 2006 | 2007.00 | 2008.0 | 2009.00 | 2010 | ▇▇▇▇▅ |
| SalePrice | 0 | 1.00 | 180921.20 | 79442.50 | 34900 | 129975.00 | 163000.0 | 214000.00 | 755000 | ▇▅▁▁▁ |
| univ\_outl\_count | 0 | 1.00 | 0.00 | 0.00 | 0 | 0.00 | 0.0 | 0.00 | 0 | ▁▁▇▁▁ |

#names(train2)

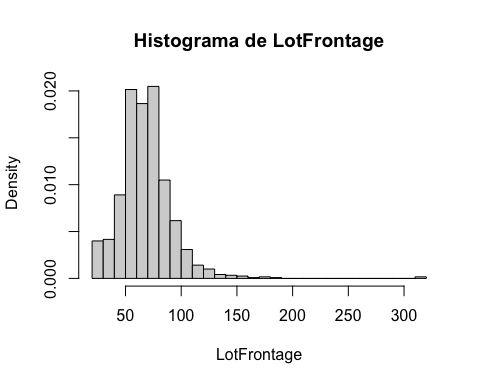
### Numerical variables

## Id (not considered for the analysis)

## “LotFrontage” Linear feet of street connected to property (cont)

numeric\_description(train2$LotFrontage, 30) # not normal distribution

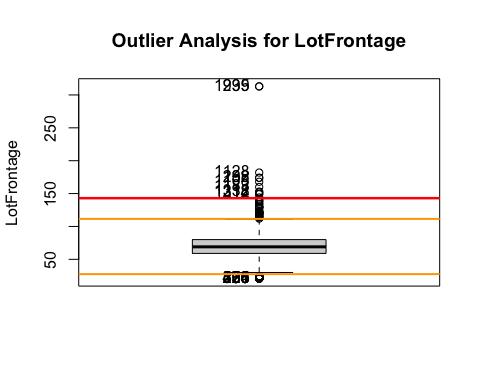
## Summary:  
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 21.00 59.00 69.00 70.05 80.00 313.00 259   
##   
## Count of missing values: 259



##   
## Shapiro-Wilk normality test  
##   
## data: variable  
## W = 0.8804, p-value < 2.2e-16

analyze\_outliers(train2, "LotFrontage")

## Number of mild outliers: 88



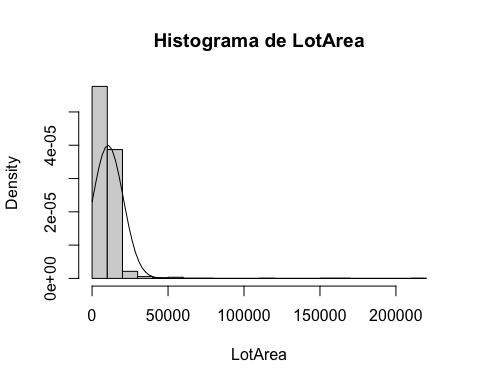
## Number of severe outliers: 12

train2 <- update\_outliers\_count(train2, "LotFrontage")

## “LotArea” Lot size in square feet (cont)

numeric\_description(train2$LotArea, 20) # not normal distribution

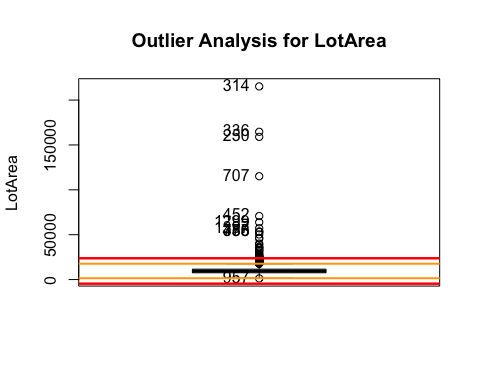
## Summary:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1300 7554 9478 10517 11602 215245   
##   
## Count of missing values: 0



##   
## Shapiro-Wilk normality test  
##   
## data: variable  
## W = 0.35106, p-value < 2.2e-16

analyze\_outliers(train2, "LotArea")

## Number of mild outliers: 69



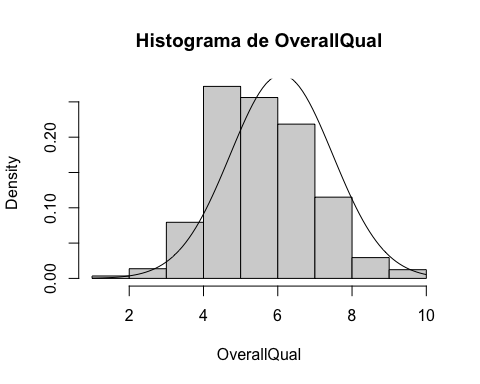
## Number of severe outliers: 34

train2 <- update\_outliers\_count(train2, "LotArea")

## “OverallQual” (cont)

numeric\_description(train2$OverallQual, 10) # not normal distribution

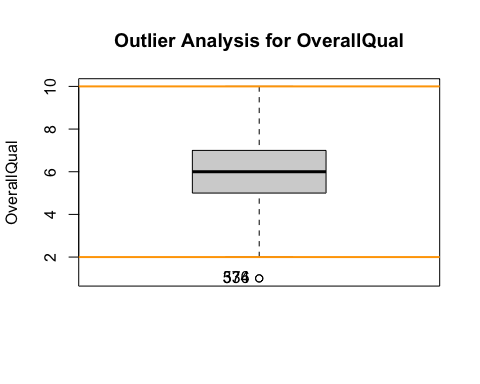
## Summary:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1.000 5.000 6.000 6.099 7.000 10.000   
##   
## Count of missing values: 0



##   
## Shapiro-Wilk normality test  
##   
## data: variable  
## W = 0.94801, p-value < 2.2e-16

analyze\_outliers(train2, "OverallQual")

## Number of mild outliers: 2



## Number of severe outliers: 0

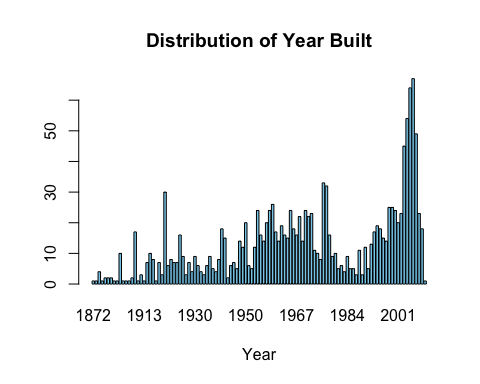
train2 <- update\_outliers\_count(train2, "OverallQual")

## “YearBuilt” (int)

summary(train2$YearBuilt)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1872 1954 1973 1971 2000 2010

barplot(table(train2$YearBuilt), main = "Distribution of Year Built",xlab = "Year",col = "skyblue")

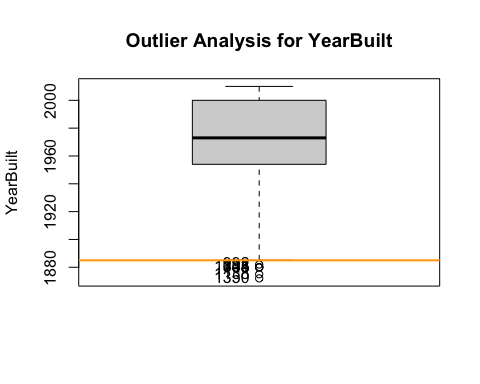


sum(is.na(train2$YearBuilt))

## [1] 0

analyze\_outliers(train2, "YearBuilt")

## Number of mild outliers: 7



## Number of severe outliers: 0

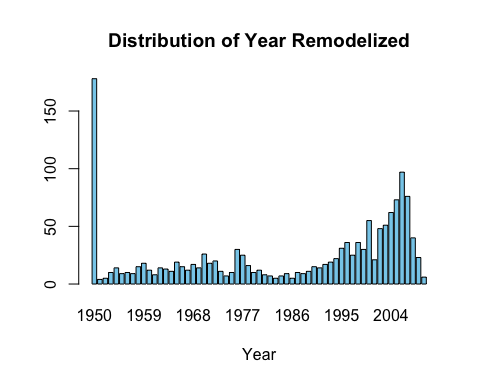
train2 <- update\_outliers\_count(train2, "YearBuilt")

## “YearRemodAdd” (int) - Remodel date (same as construction date if no remodeling or additions)

summary(train2$YearRemodAdd)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1950 1967 1994 1985 2004 2010

barplot(table(train2$YearRemodAdd), main = "Distribution of Year Remodelized",xlab = "Year",col = "skyblue")

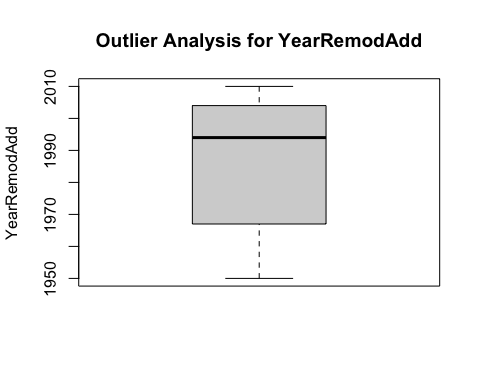


sum(is.na(train2$YearRemodAdd))

## [1] 0

analyze\_outliers(train2, "YearRemodAdd")

## Number of mild outliers: 0



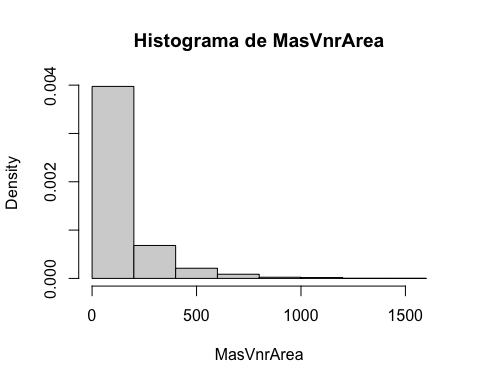
## Number of severe outliers: 0

train2 <- update\_outliers\_count(train2, "YearRemodAdd")

## MasVnrArea - Masonry veneer area in square feet (cont)

numeric\_description(train2$MasVnrArea, 10)# not normal distribution

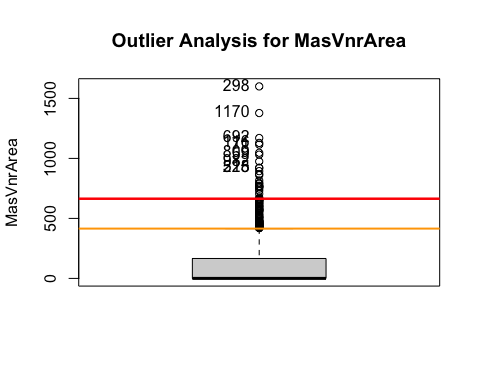
## Summary:  
## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.0 0.0 0.0 103.7 166.0 1600.0 8   
##   
## Count of missing values: 8



##   
## Shapiro-Wilk normality test  
##   
## data: variable  
## W = 0.63929, p-value < 2.2e-16

analyze\_outliers(train2,"MasVnrArea")

## Number of mild outliers: 96



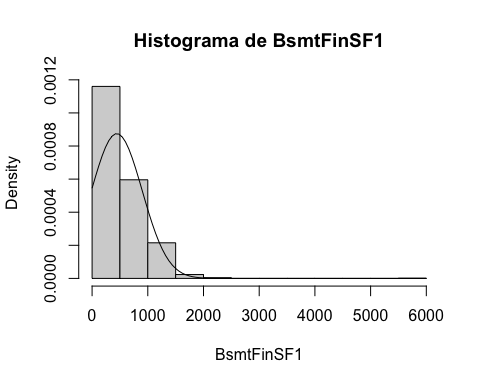
## Number of severe outliers: 25

train2 <- update\_outliers\_count(train2, "MasVnrArea")  
train2$MasVnrArea[which(train2$MasVnrArea == 0)] <- NA  
none\_idx <- which(train$MasVnrType == 'None')  
train2$MasVnrArea[none\_idx] <- 0

## “BsmtFinSF1” - Type 1 finished square feet (cont)

numeric\_description(train2$BsmtFinSF1, 10)# not normal distribution

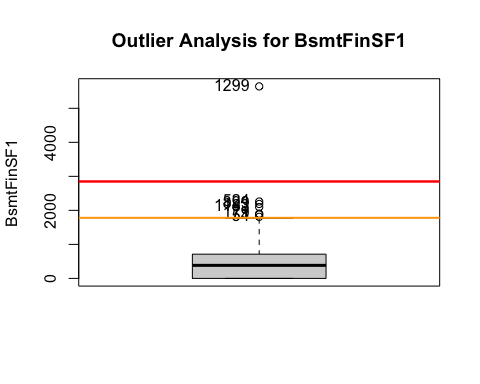
## Summary:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0 0.0 383.5 443.6 712.2 5644.0   
##   
## Count of missing values: 0



##   
## Shapiro-Wilk normality test  
##   
## data: variable  
## W = 0.84796, p-value < 2.2e-16

analyze\_outliers(train2,"BsmtFinSF1")

## Number of mild outliers: 7



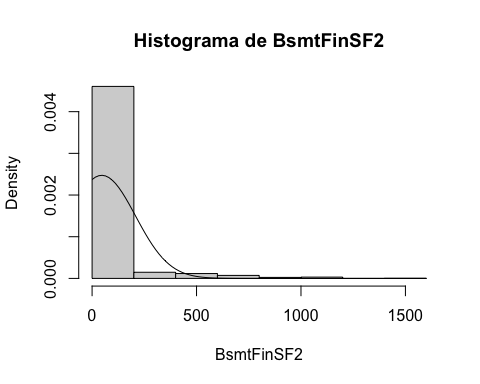
## Number of severe outliers: 1

train2 <- update\_outliers\_count(train2, "BsmtFinSF1")

# BsmtFinSF2 - Rating of basement finished area (if multiple types) (cont)

numeric\_description(train2$BsmtFinSF2, 10) # not normal distribution

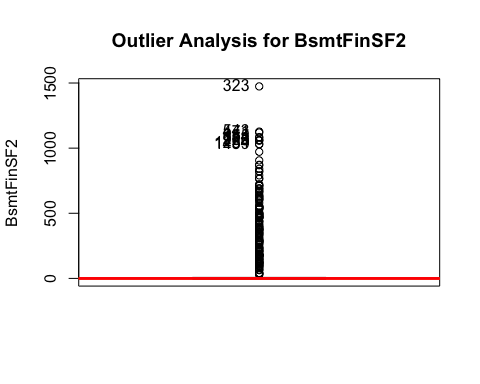
## Summary:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 0.00 0.00 46.55 0.00 1474.00   
##   
## Count of missing values: 0



##   
## Shapiro-Wilk normality test  
##   
## data: variable  
## W = 0.32728, p-value < 2.2e-16

analyze\_outliers(train2,"BsmtFinSF2")

## Number of mild outliers: 167



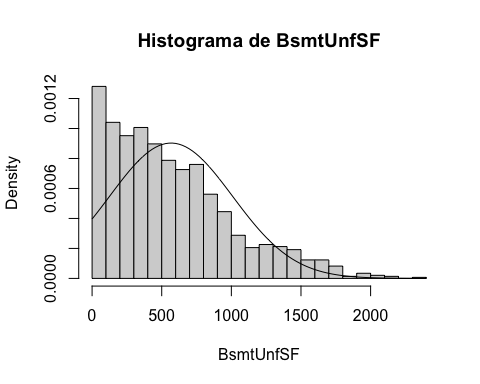
## Number of severe outliers: 167

train2 <- update\_outliers\_count(train2, "BsmtFinSF2")

# “BsmtUnfSF” (cont)

numeric\_description(train2$BsmtUnfSF, 20) # not normal distribution

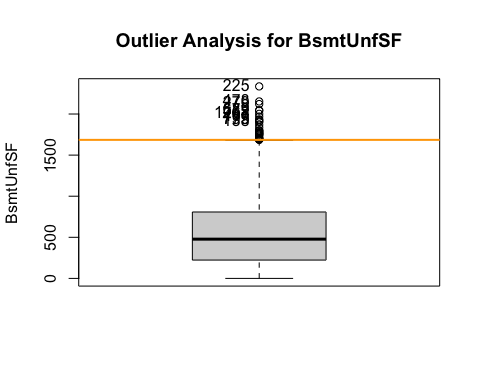
## Summary:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0 223.0 477.5 567.2 808.0 2336.0   
##   
## Count of missing values: 0



##   
## Shapiro-Wilk normality test  
##   
## data: variable  
## W = 0.93042, p-value < 2.2e-16

analyze\_outliers(train2,"BsmtUnfSF")

## Number of mild outliers: 29



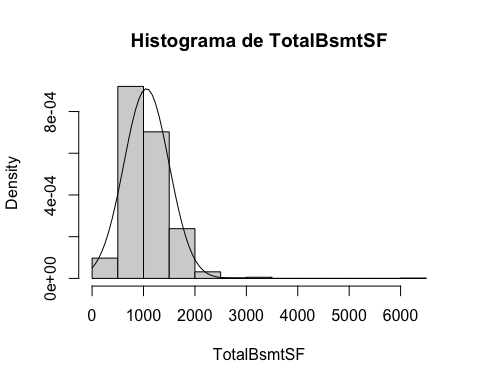
## Number of severe outliers: 0

train2 <- update\_outliers\_count(train2, "BsmtUnfSF")

# “TotalBsmtSF” (cont)

numeric\_description(train2$TotalBsmtSF, 10)# not normal distribution

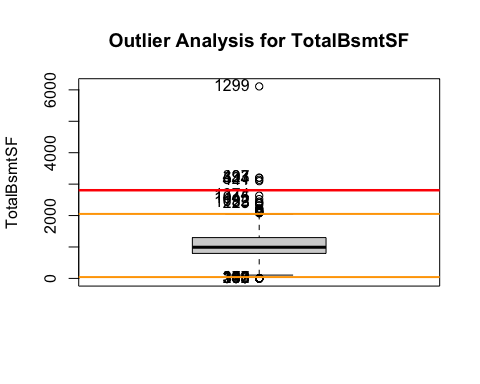
## Summary:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0 795.8 991.5 1057.4 1298.2 6110.0   
##   
## Count of missing values: 0



##   
## Shapiro-Wilk normality test  
##   
## data: variable  
## W = 0.91735, p-value < 2.2e-16

analyze\_outliers(train2,"TotalBsmtSF")

## Number of mild outliers: 61



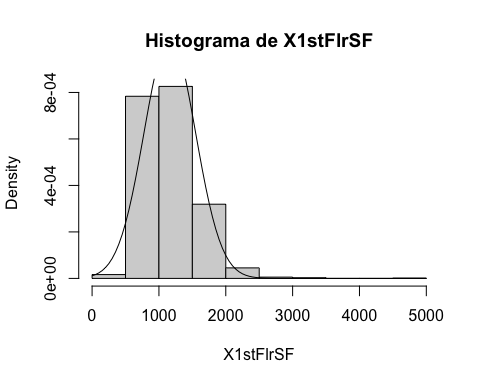
## Number of severe outliers: 5

train2 <- update\_outliers\_count(train2, "TotalBsmtSF")

# X1stFlrSF - First Floor square feet (cont)

numeric\_description(train2$X1stFlrSF, 10)

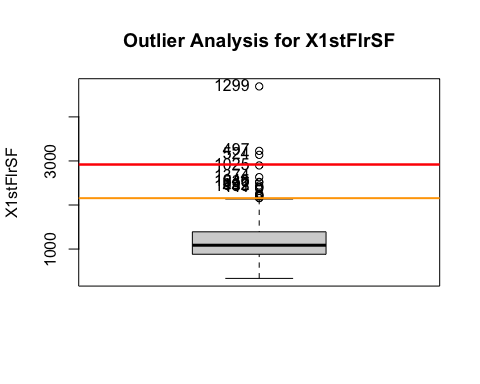
## Summary:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 334 882 1087 1163 1391 4692   
##   
## Count of missing values: 0



##   
## Shapiro-Wilk normality test  
##   
## data: variable  
## W = 0.92695, p-value < 2.2e-16

# not normal distribution  
analyze\_outliers(train2,"X1stFlrSF")

## Number of mild outliers: 20



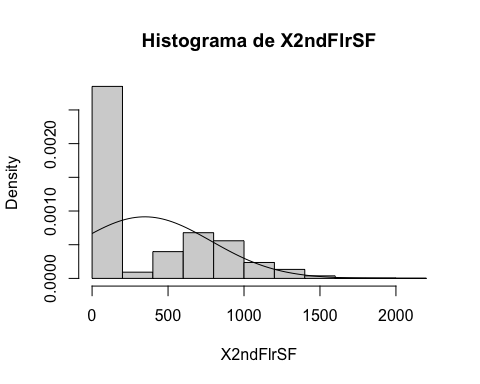
## Number of severe outliers: 3

train2 <- update\_outliers\_count(train2, "X1stFlrSF")

## X2ndFlrSF - Second floor square feet (cont)

numeric\_description(train2$X2ndFlrSF, 10)# not normal distribution

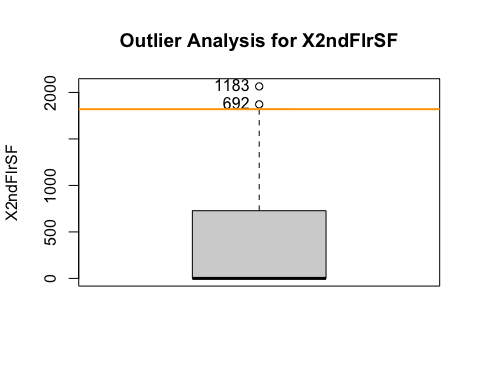
## Summary:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0 0 0 347 728 2065   
##   
## Count of missing values: 0



##   
## Shapiro-Wilk normality test  
##   
## data: variable  
## W = 0.7668, p-value < 2.2e-16

analyze\_outliers(train2,"X2ndFlrSF")

## Number of mild outliers: 2



## Number of severe outliers: 0

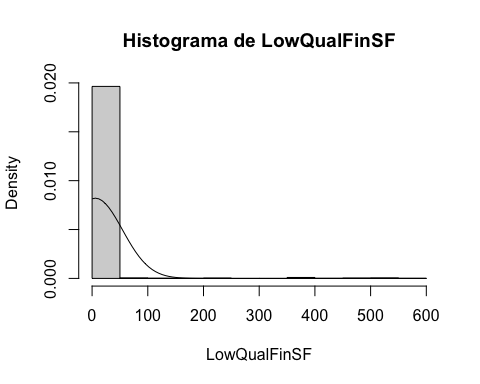
train2 <- update\_outliers\_count(train2, "X2ndFlrSF")  
  
# create a new variable if has second floor = 1 / no second floor = 0.   
train2$secondfloor<-0  
train2$secondfloor[which(train2$X2ndFlrSF!=0)]<-1  
train2$secondfloor<-factor(train2$secondfloor)  
table(train2$secondfloor)

##   
## 0 1   
## 829 631

## LowQualFinSF - Low quality finished square feet (all floors) (cont)

numeric\_description(train2$LowQualFinSF, 10) # not normal distribution

## Summary:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 0.000 0.000 5.845 0.000 572.000   
##   
## Count of missing values: 0



##   
## Shapiro-Wilk normality test  
##   
## data: variable  
## W = 0.09799, p-value < 2.2e-16

table(train2$LowQualFinSF)# very centered in 0

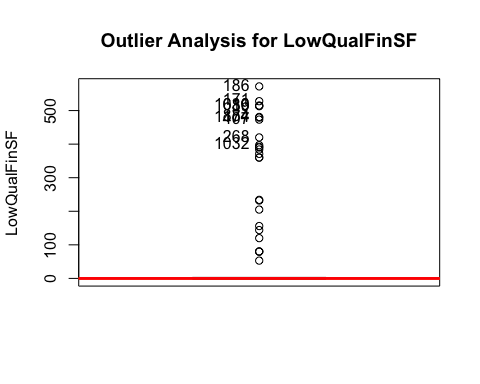
##   
## 0 53 80 120 144 156 205 232 234 360 371 384 390 392 397 420   
## 1434 1 3 1 1 1 1 1 1 2 1 1 1 1 1 1   
## 473 479 481 513 514 515 528 572   
## 1 1 1 1 1 1 1 1

nearZeroVar(train2$LowQualFinSF, saveMetrics = TRUE) # Variable with near zero variance

## freqRatio percentUnique zeroVar nzv  
## 1 478 1.643836 FALSE TRUE

analyze\_outliers(train2,"LowQualFinSF")

## Number of mild outliers: 26



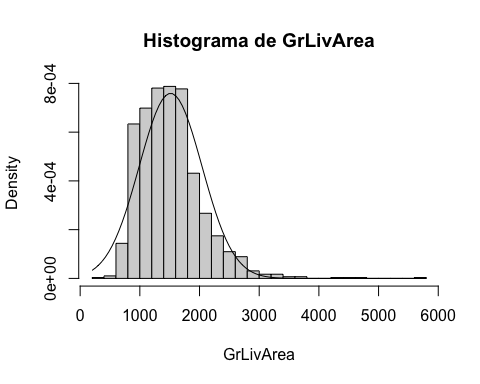
## Number of severe outliers: 26

train2 <- update\_outliers\_count(train2, "LowQualFinSF")

## GrLivArea - Above grade (ground) living area square feet (cont)

numeric\_description(train2$GrLivArea, 20) # not normal distribution

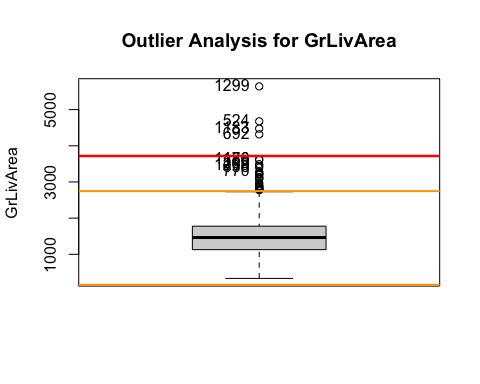
## Summary:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 334 1130 1464 1515 1777 5642   
##   
## Count of missing values: 0



##   
## Shapiro-Wilk normality test  
##   
## data: variable  
## W = 0.92798, p-value < 2.2e-16

analyze\_outliers(train2,"GrLivArea")

## Number of mild outliers: 31



## Number of severe outliers: 4

train2 <- update\_outliers\_count(train2, "GrLivArea")

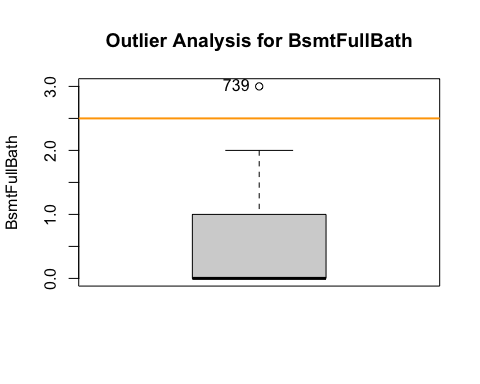
## BsmtFullBath - # full baths in the basement (int)

summary(train2$BsmtFullBath)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 0.0000 0.4253 1.0000 3.0000

analyze\_outliers(train2,"BsmtFullBath")

## Number of mild outliers: 1



## Number of severe outliers: 0

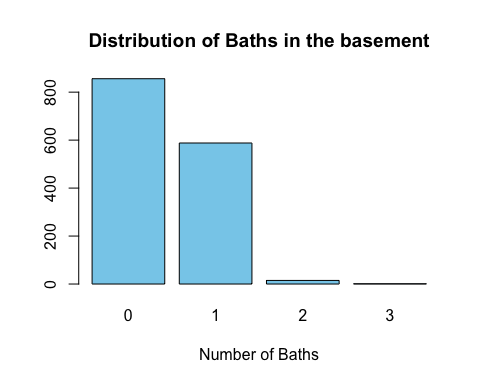
train2 <- update\_outliers\_count(train2, "BsmtFullBath")  
table(train2$BsmtFullBath)

##   
## 0 1 2 3   
## 856 588 15 1

sum(is.na(train2$BsmtFullBath))

## [1] 0

barplot(table(train2$BsmtFullBath), main = "Distribution of Baths in the basement",xlab = "Number of Baths",col = "skyblue")



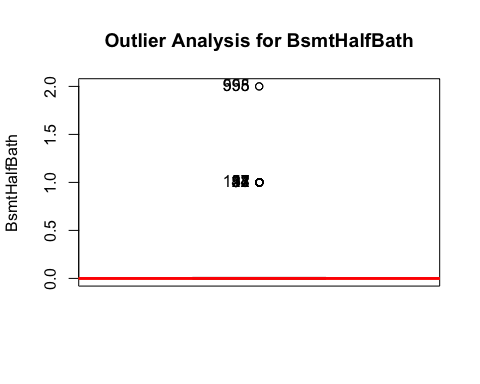
## BsmtHalfBath - # half baths in the basement (int)

summary(train2$BsmtHalfBath)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00000 0.00000 0.00000 0.05753 0.00000 2.00000

analyze\_outliers(train2,"BsmtHalfBath")

## Number of mild outliers: 82



## Number of severe outliers: 82

train2 <- update\_outliers\_count(train2, "BsmtHalfBath")  
table(train2$BsmtHalfBath) # very centered in 0

##   
## 0 1 2   
## 1378 80 2

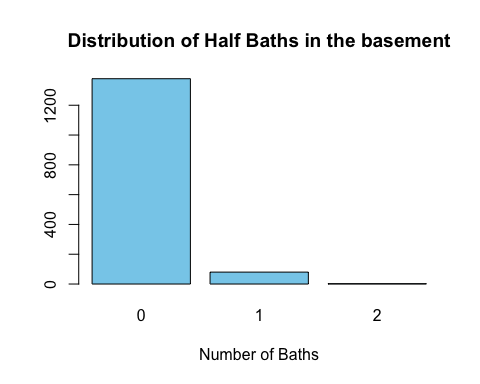
nearZeroVar(train2$BsmtHalfBath, saveMetrics = TRUE) # Variable not with near zero variance

## freqRatio percentUnique zeroVar nzv  
## 1 17.225 0.2054795 FALSE FALSE

sum(is.na(train2$BsmtHalfBath))

## [1] 0

barplot(table(train2$BsmtHalfBath), main = "Distribution of Half Baths in the basement",xlab = "Number of Baths",col = "skyblue")



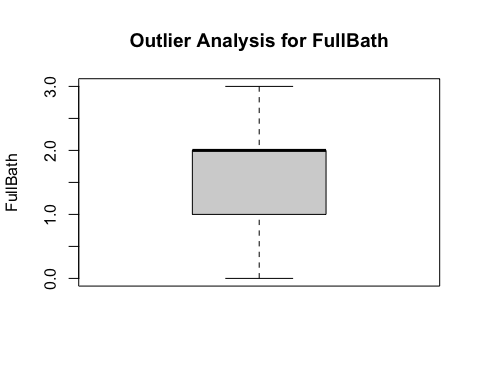
## FullBath - full baths above grade (int)

summary(train2$FullBath)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 1.000 2.000 1.565 2.000 3.000

analyze\_outliers(train2,"FullBath")

## Number of mild outliers: 0



## Number of severe outliers: 0

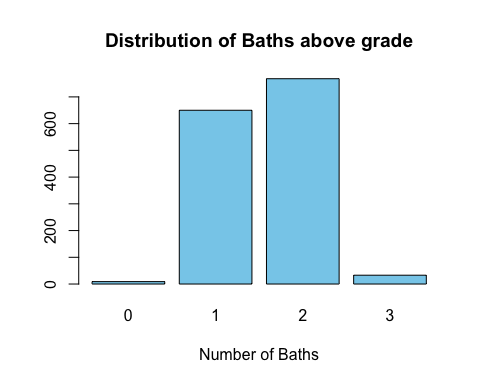
train2 <- update\_outliers\_count(train2, "FullBath")  
table(train2$FullBath)

##   
## 0 1 2 3   
## 9 650 768 33

sum(is.na(train2$FullBath))

## [1] 0

barplot(table(train2$FullBath), main = "Distribution of Baths above grade",xlab = "Number of Baths",col = "skyblue")



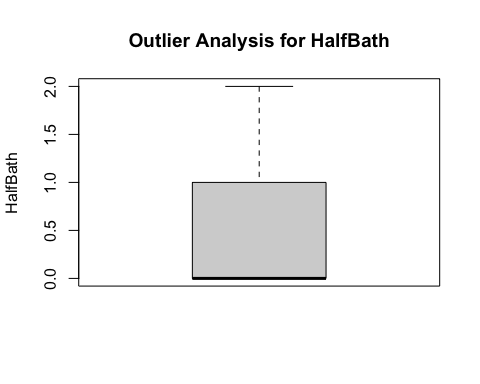
## HalfBath - Half baths above grade (int)

summary(train2$HalfBath)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0000 0.0000 0.0000 0.3829 1.0000 2.0000

analyze\_outliers(train2,"HalfBath")

## Number of mild outliers: 0



## Number of severe outliers: 0

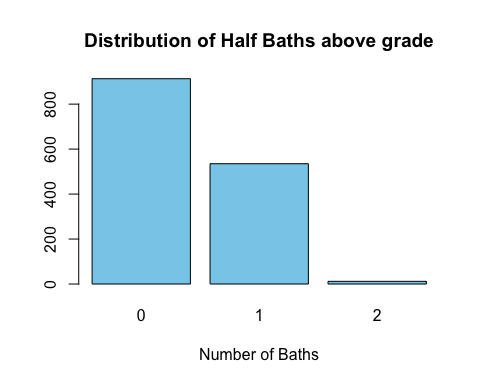
train2 <- update\_outliers\_count(train2, "HalfBath")  
table(train2$HalfBath)

##   
## 0 1 2   
## 913 535 12

sum(is.na(train2$HalfBath))

## [1] 0

barplot(table(train2$HalfBath), main = "Distribution of Half Baths above grade",xlab = "Number of Baths",col = "skyblue")



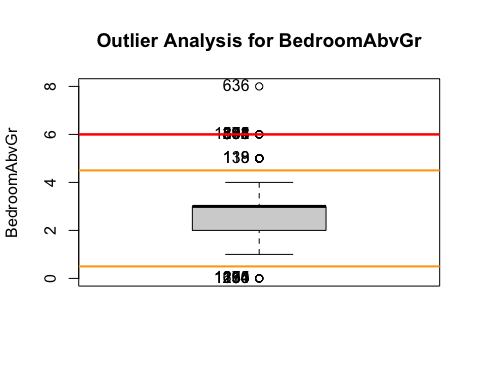
## BedroomAbvGr - Bedrooms above grade (does NOT include basement bedrooms)

summary(train2$BedroomAbvGr)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 2.000 3.000 2.866 3.000 8.000

analyze\_outliers(train2,"BedroomAbvGr")

## Number of mild outliers: 35



## Number of severe outliers: 1

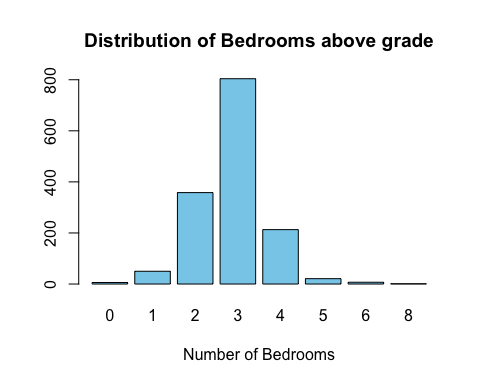
train2 <- update\_outliers\_count(train2, "BedroomAbvGr")  
table(train2$BedroomAbvGr)

##   
## 0 1 2 3 4 5 6 8   
## 6 50 358 804 213 21 7 1

sum(is.na(train2$BedroomAbvGr))

## [1] 0

barplot(table(train2$BedroomAbvGr), main = "Distribution of Bedrooms above grade",xlab = "Number of Bedrooms",col = "skyblue")



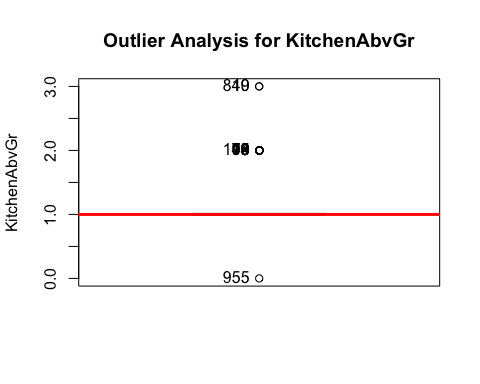
## KitchenAbvGr - Kitchens above grade (int)

summary(train2$KitchenAbvGr)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 1.000 1.000 1.047 1.000 3.000

analyze\_outliers(train2,"KitchenAbvGr")

## Number of mild outliers: 68



## Number of severe outliers: 68

train2 <- update\_outliers\_count(train2, "KitchenAbvGr")  
table(train2$KitchenAbvGr) # variable very centered in 1

##   
## 0 1 2 3   
## 1 1392 65 2

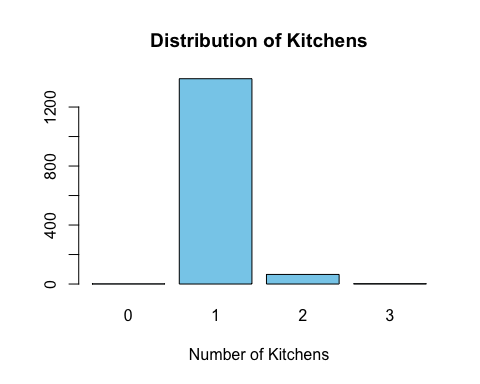
nearZeroVar(train2$KitchenAbvGr,saveMetrics = TRUE) # near zero variance

## freqRatio percentUnique zeroVar nzv  
## 1 21.41538 0.2739726 FALSE TRUE

sum(is.na(train2$KitchenAbvGr))

## [1] 0

barplot(table(train2$KitchenAbvGr), main = "Distribution of Kitchens",xlab = "Number of Kitchens",col = "skyblue")



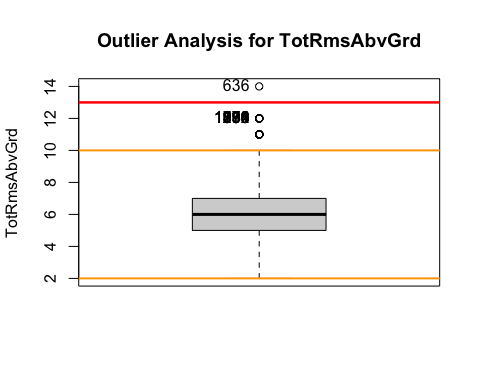
## “TotRmsAbvGrd” - Total rooms above grade (does not include bathrooms) - continuous ratio variable (int)

summary(train2$TotRmsAbvGrd)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 2.000 5.000 6.000 6.518 7.000 14.000

analyze\_outliers(train2,"TotRmsAbvGrd")

## Number of mild outliers: 30



## Number of severe outliers: 1

train2 <- update\_outliers\_count(train2, "TotRmsAbvGrd")  
#table(train2$TotRmsAbvGrd)  
sum(is.na(train2$TotRmsAbvGrd))

## [1] 0

## “Fireplaces” - Number of fireplaces (int)

summary(train2$Fireplaces)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 0.000 1.000 0.613 1.000 3.000

table(train2$Fireplaces)

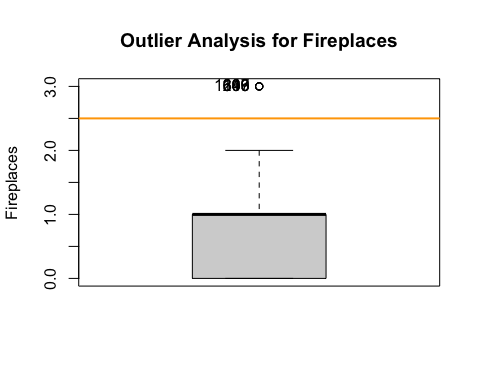
##   
## 0 1 2 3   
## 690 650 115 5

sum(is.na(train2$Fireplaces)) #0

## [1] 0

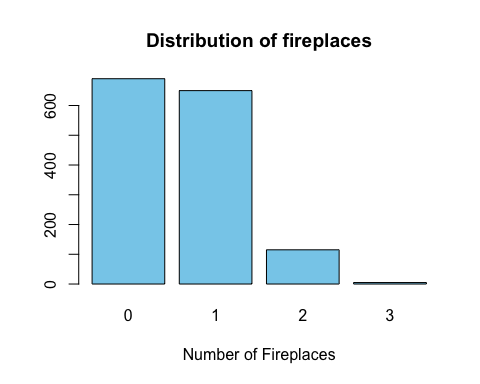
analyze\_outliers(train2, "Fireplaces")

## Number of mild outliers: 5



## Number of severe outliers: 0

train2 <- update\_outliers\_count(train2, "Fireplaces")  
barplot(table(train2$Fireplaces), main = "Distribution of fireplaces",xlab = "Number of Fireplaces",col = "skyblue")



## “GarageCars” - Size of garage in car capacity (int)

summary(train2$GarageCars)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 1.000 2.000 1.767 2.000 4.000

table(train2$GarageCars)

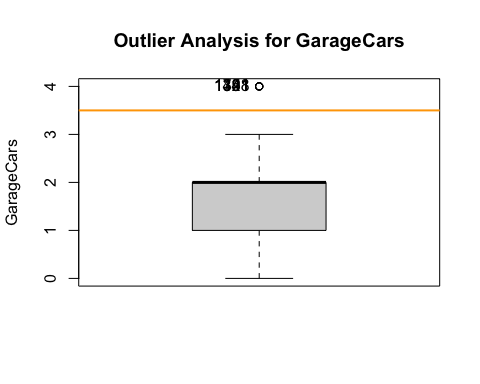
##   
## 0 1 2 3 4   
## 81 369 824 181 5

sum(is.na(train2$GarageCars)) #0

## [1] 0

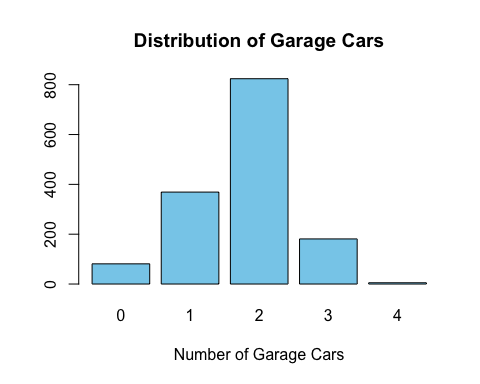
analyze\_outliers(train2, "GarageCars")

## Number of mild outliers: 5



## Number of severe outliers: 0

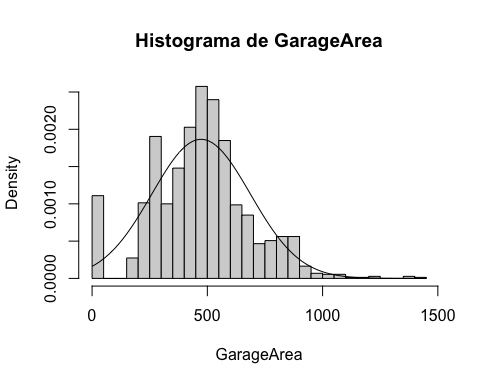
train2 <- update\_outliers\_count(train2, "GarageCars")  
barplot(table(train2$GarageCars), main = "Distribution of Garage Cars",xlab = "Number of Garage Cars",col = "skyblue")



## “GarageArea” - Size of garage in square feet (cont)

numeric\_description(train2$GarageArea, 30)

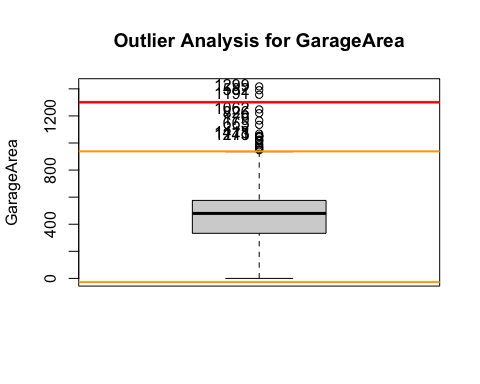
## Summary:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0 334.5 480.0 473.0 576.0 1418.0   
##   
## Count of missing values: 0



##   
## Shapiro-Wilk normality test  
##   
## data: variable  
## W = 0.97533, p-value = 4.017e-15

# not normal distribution  
analyze\_outliers(train2, "GarageArea")

## Number of mild outliers: 21



## Number of severe outliers: 3

train2 <- update\_outliers\_count(train2, "GarageArea")  
sm<-summary(train2$GarageArea)  
  
# checking if 0 mean that they don't have garage  
which(train$GarageCars == 0)

## [1] 40 49 79 89 90 100 109 126 128 141 149 156 164 166 199  
## [16] 211 242 251 288 292 308 376 387 394 432 435 442 465 496 521  
## [31] 529 534 536 563 583 614 615 621 636 637 639 650 706 711 739  
## [46] 751 785 827 844 922 943 955 961 969 971 977 1010 1012 1031 1039  
## [61] 1097 1124 1132 1138 1144 1174 1180 1219 1220 1235 1258 1284 1324 1326 1327  
## [76] 1338 1350 1408 1450 1451 1454

which(train$GarageArea == 0)

## [1] 40 49 79 89 90 100 109 126 128 141 149 156 164 166 199  
## [16] 211 242 251 288 292 308 376 387 394 432 435 442 465 496 521  
## [31] 529 534 536 563 583 614 615 621 636 637 639 650 706 711 739  
## [46] 751 785 827 844 922 943 955 961 969 971 977 1010 1012 1031 1039  
## [61] 1097 1124 1132 1138 1144 1174 1180 1219 1220 1235 1258 1284 1324 1326 1327  
## [76] 1338 1350 1408 1450 1451 1454

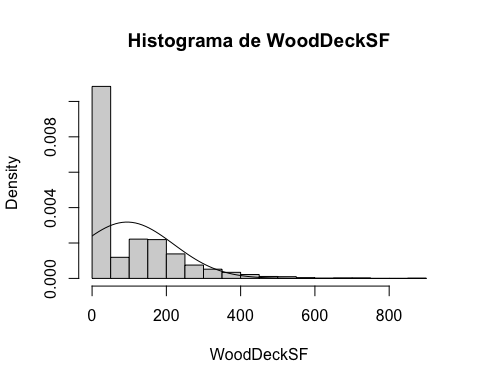
which(train$GarageFinish=='No Garage')

## [1] 40 49 79 89 90 100 109 126 128 141 149 156 164 166 199  
## [16] 211 242 251 288 292 308 376 387 394 432 435 442 465 496 521  
## [31] 529 534 536 563 583 614 615 621 636 637 639 650 706 711 739  
## [46] 751 785 827 844 922 943 955 961 969 971 977 1010 1012 1031 1039  
## [61] 1097 1124 1132 1138 1144 1174 1180 1219 1220 1235 1258 1284 1324 1326 1327  
## [76] 1338 1350 1408 1450 1451 1454

## “WoodDeckSF” - Wood deck area in square feet - continuous ratio variable

numeric\_description(train2$WoodDeckSF, 30) # not normal distribution

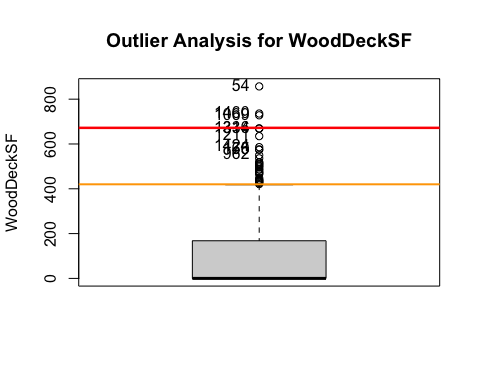
## Summary:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 0.00 0.00 94.24 168.00 857.00   
##   
## Count of missing values: 0



##   
## Shapiro-Wilk normality test  
##   
## data: variable  
## W = 0.76852, p-value < 2.2e-16

#table(train2$WoodDeckSF) # centered data in 0  
analyze\_outliers(train2, "WoodDeckSF")

## Number of mild outliers: 32



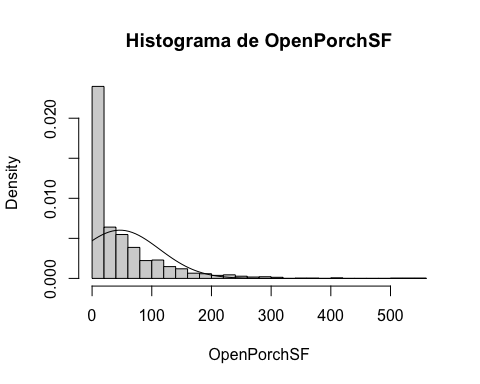
## Number of severe outliers: 3

train2 <- update\_outliers\_count(train2, "WoodDeckSF")  
sm<-summary(train2$WoodDeckSF)

## “OpenPorchSF” - Open porch area in square feet - continuous ratio variable (continua)

numeric\_description(train2$OpenPorchSF, 30)

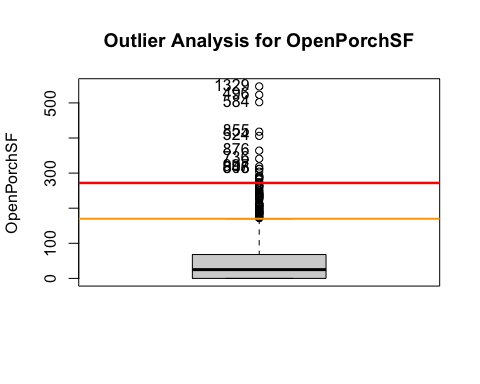
## Summary:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 0.00 25.00 46.66 68.00 547.00   
##   
## Count of missing values: 0



##   
## Shapiro-Wilk normality test  
##   
## data: variable  
## W = 0.72717, p-value < 2.2e-16

analyze\_outliers(train2,"OpenPorchSF")

## Number of mild outliers: 77



## Number of severe outliers: 18

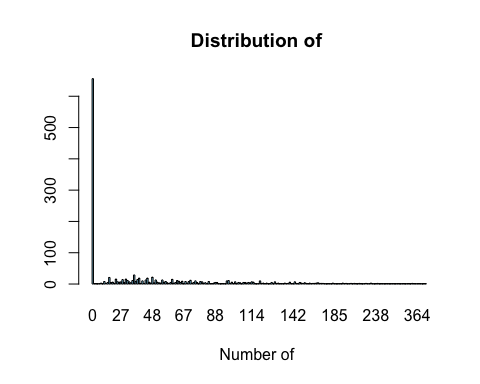
train2 <- update\_outliers\_count(train2, "OpenPorchSF")  
summary(train2$OpenPorchSF)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 0.00 25.00 46.66 68.00 547.00

#table(train2$OpenPorchSF) # very centered data in 0  
sum(is.na(train2$OpenPorchSF))

## [1] 0

barplot(table(train2$OpenPorchSF), main = "Distribution of ",xlab = "Number of ",col = "skyblue")



sm<-summary(train2$OpenPorchSF)  
train2$OpenPorch\_binary<-0  
train2$OpenPorch\_binary[which(train2$OpenPorchSF!=0)]<-1  
train2$OpenPorch\_binary<-as.factor(train2$OpenPorch\_binary)  
table(train2$OpenPorch\_binary)

##   
## 0 1   
## 656 804

## EnclosedPorch - Enclosed porch area in square feet (cont)

summary(train2$EnclosedPorch)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 0.00 0.00 21.95 0.00 552.00

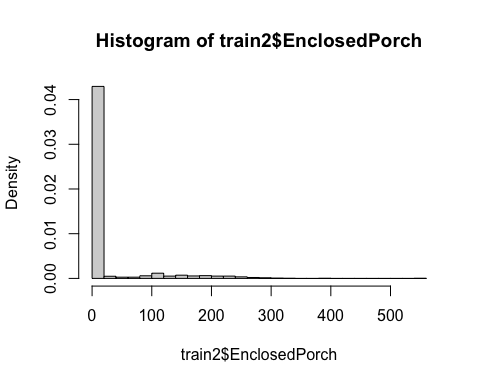
table(train2$EnclosedPorch) # very centered data in 0

##   
## 0 19 20 24 30 32 34 36 37 39 40 42 44 48 50 52   
## 1252 1 1 1 1 2 2 2 1 2 3 1 1 1 1 1   
## 54 56 60 64 67 68 70 77 80 81 84 87 90 91 94 96   
## 1 1 1 1 1 1 1 2 2 1 2 1 1 1 1 6   
## 98 99 100 102 105 108 112 114 115 116 120 123 126 128 129 130   
## 1 1 2 3 1 2 15 2 2 4 5 1 3 3 1 1   
## 134 136 137 138 140 143 144 145 148 150 154 156 158 160 162 164   
## 1 1 2 1 1 1 5 1 1 3 2 4 2 2 1 3   
## 168 169 170 172 174 176 177 180 183 184 185 189 190 192 194 196   
## 2 1 1 1 1 3 1 2 1 3 2 1 2 5 1 1   
## 198 200 202 205 208 210 212 214 216 218 220 221 224 226 228 230   
## 1 1 2 2 1 1 1 1 5 1 1 1 2 1 3 1   
## 234 236 239 240 242 244 248 252 254 259 264 268 272 275 280 286   
## 2 2 1 2 1 2 1 4 1 1 2 1 1 1 1 1   
## 291 293 294 301 318 330 386 552   
## 1 1 1 1 1 1 1 1

sum(is.na(train2$EnclosedPorch))

## [1] 0

hist(train2$EnclosedPorch, breaks = 30, freq = F)



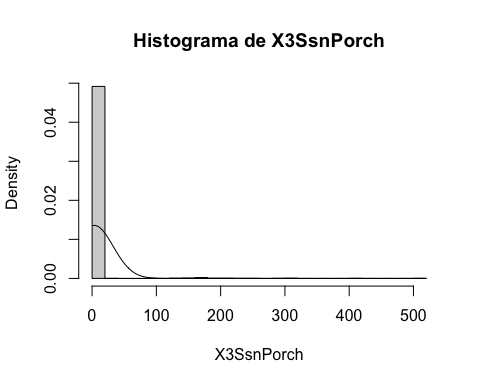
# therefore we will create a factor 0 if they dont have enclosed porch area,1 if they do have.  
train2$EnclosedPorch\_binary<-0  
train2$EnclosedPorch\_binary[which(train2$EnclosedPorch!=0)]<-1  
train2$EnclosedPorch\_binary<-as.factor(train2$EnclosedPorch\_binary)  
#table(train2$EnclosedPorch\_binary)  
  
# create new variable - has porch or not - mixing both binary variables  
train2$HasPorch\_binary<-0  
train2$HasPorch\_binary[which(train2$EnclosedPorch!=0 | train2$OpenPorchSF!=0)]<-1  
train2$HasPorch\_binary<-as.factor(train2$HasPorch\_binary)  
table(train2$HasPorch\_binary)

##   
## 0 1   
## 513 947

## X3SsnPorch - Three season porch area in square feet (cont)

numeric\_description(train2$X3SsnPorch, 30)

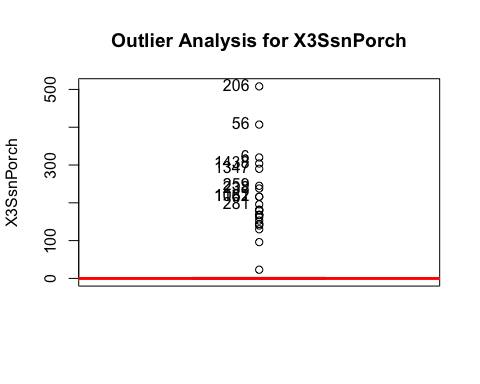
## Summary:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 0.00 0.00 3.41 0.00 508.00   
##   
## Count of missing values: 0



##   
## Shapiro-Wilk normality test  
##   
## data: variable  
## W = 0.094934, p-value < 2.2e-16

#table(train2$X3SsnPorch) # very centered in 0 -> dont have three season porch  
analyze\_outliers(train2,"X3SsnPorch")

## Number of mild outliers: 24



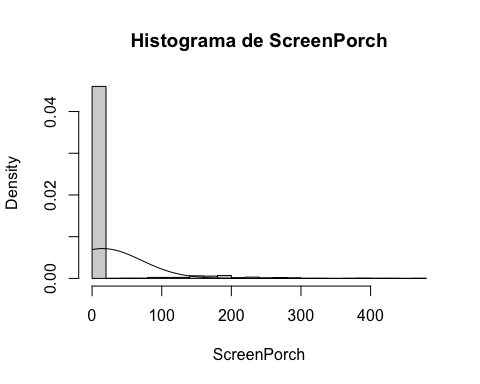
## Number of severe outliers: 24

train2 <- update\_outliers\_count(train2, "X3SsnPorch")

## ScreenPorch - Screen porch area in square feet (cont)

numeric\_description(train2$ScreenPorch, 30) # not normally distributed

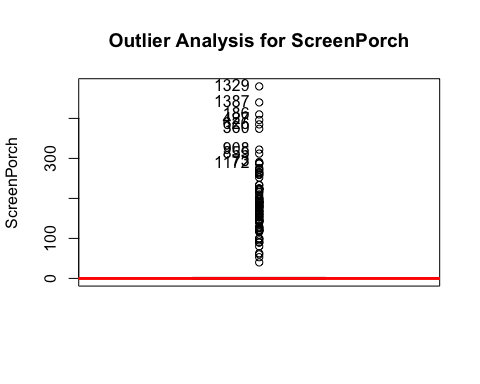
## Summary:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 0.00 0.00 15.06 0.00 480.00   
##   
## Count of missing values: 0



##   
## Shapiro-Wilk normality test  
##   
## data: variable  
## W = 0.29821, p-value < 2.2e-16

#table(train2$ScreenPorch) # very centered in 0 -> dont have screen porch  
analyze\_outliers(train2,"ScreenPorch")

## Number of mild outliers: 116



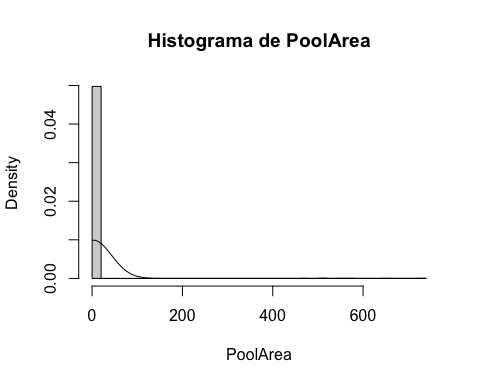
## Number of severe outliers: 116

train2 <- update\_outliers\_count(train2, "ScreenPorch")

## PoolArea - Pool area in square feet (cont)

numeric\_description(train2$PoolArea, 30)

## Summary:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.000 0.000 0.000 2.759 0.000 738.000   
##   
## Count of missing values: 0



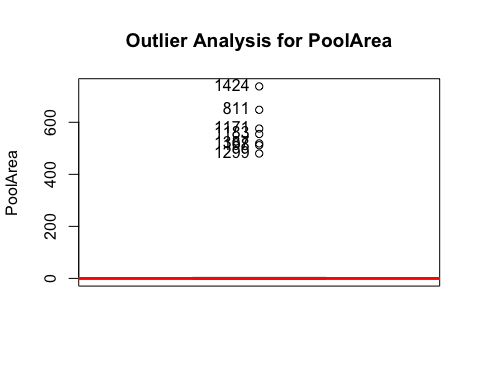
##   
## Shapiro-Wilk normality test  
##   
## data: variable  
## W = 0.041202, p-value < 2.2e-16

# not normally distributed  
table(train2$PoolArea) # very centered in 0 -> dont have pool

##   
## 0 480 512 519 555 576 648 738   
## 1453 1 1 1 1 1 1 1

analyze\_outliers(train2,"PoolArea")

## Number of mild outliers: 7



## Number of severe outliers: 7

train2 <- update\_outliers\_count(train2, "PoolArea")  
# create a new variable : 0=no pool/1= pool  
train2$Pool\_binary<-0  
train2$Pool\_binary[which(train2$PoolArea!=0)]<-1  
train2$Pool\_binary<-as.factor(train2$Pool\_binary)  
table(train2$Pool\_binary) # not very usefull , the majority dont have pools, very centered

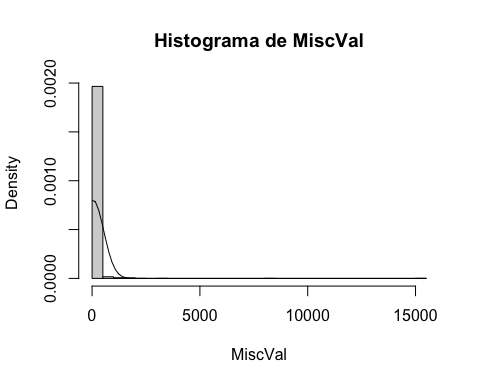
##   
## 0 1   
## 1453 7

#train2$Pool\_binary<-NULL

## MiscVal - $Value of miscellaneous feature (cont)

numeric\_description(train2$MiscVal, 30)

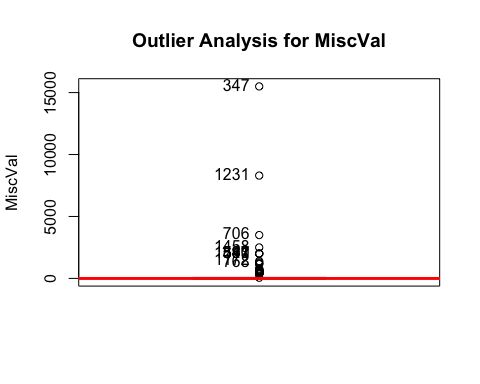
## Summary:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 0.00 0.00 43.49 0.00 15500.00   
##   
## Count of missing values: 0



##   
## Shapiro-Wilk normality test  
##   
## data: variable  
## W = 0.058233, p-value < 2.2e-16

# not normally distributed  
#table(train2$MiscVal) # very centered in 0   
analyze\_outliers(train2,"MiscVal")

## Number of mild outliers: 52

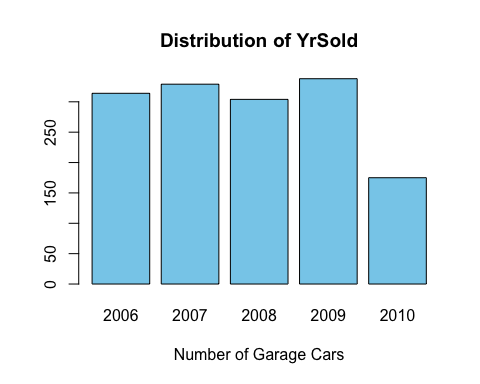


## Number of severe outliers: 52

train2 <- update\_outliers\_count(train2, "MiscVal")

## YrSold - (int)

barplot(table(train2$YrSold), main = "Distribution of YrSold",xlab = "Number of Garage Cars",col = "skyblue")

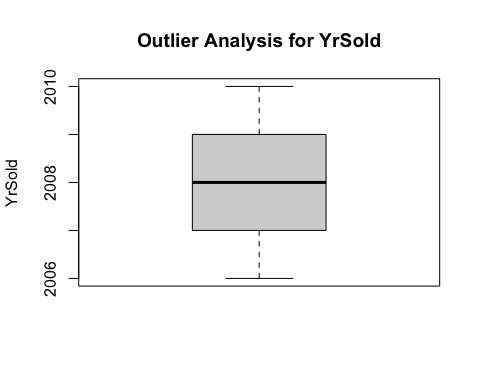


table(train2$YrSold)

##   
## 2006 2007 2008 2009 2010   
## 314 329 304 338 175

analyze\_outliers(train2,"YrSold")

## Number of mild outliers: 0



## Number of severe outliers: 0

train2 <- update\_outliers\_count(train2, "YrSold")

### Categorical Variables

## Neighborhood

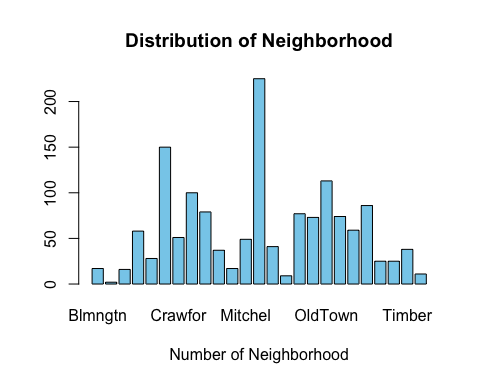
sum(is.na(train2$Neighborhood)) #0

## [1] 0

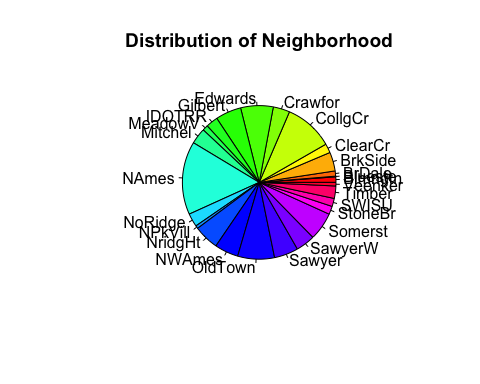
table(train2$Neighborhood)

##   
## Blmngtn Blueste BrDale BrkSide ClearCr CollgCr Crawfor Edwards Gilbert IDOTRR   
## 17 2 16 58 28 150 51 100 79 37   
## MeadowV Mitchel NAmes NoRidge NPkVill NridgHt NWAmes OldTown Sawyer SawyerW   
## 17 49 225 41 9 77 73 113 74 59   
## Somerst StoneBr SWISU Timber Veenker   
## 86 25 25 38 11

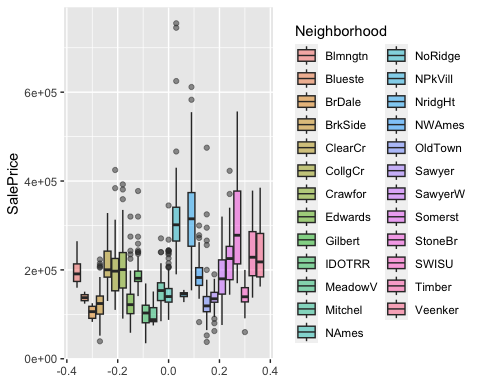
barplot(table(train2$Neighborhood), main = "Distribution of Neighborhood",xlab = "Number of Neighborhood",col = "skyblue")



pie(table(train2$Neighborhood), main = "Distribution of Neighborhood", col = rainbow(length(levels(train2$Neighborhood))))



# analysis with the target variable  
#ggplot(train2, aes(x=SalePrice, fill=Neighborhood)) +geom\_density(alpha=.5)  
ggplot(train2, aes(y=SalePrice, fill=Neighborhood)) +  
 geom\_boxplot(alpha=0.5)



#pairwise.wilcox.test(train2$SalePrice, train2$Neighborhood)

## ExterQual - Evaluates the quality of the material on the exterior

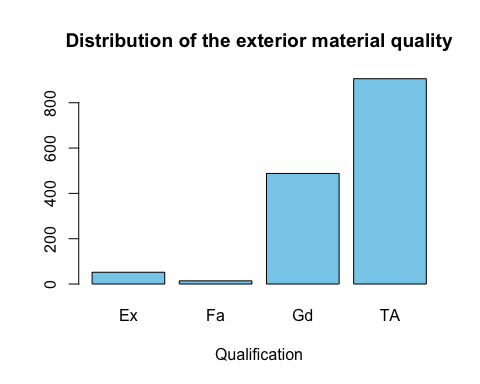
sum(is.na(train2$ExterQual)) #0

## [1] 0

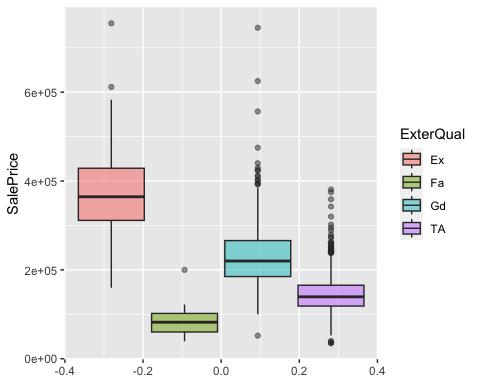
table(train2$ExterQual)

##   
## Ex Fa Gd TA   
## 52 14 488 906

barplot(table(train2$ExterQual), main = "Distribution of the exterior material quality",xlab = "Qualification",col = "skyblue")



# analysis with the target variable  
#ggplot(train2, aes(x=SalePrice, fill=ExterQual)) +geom\_density(alpha=.5)  
ggplot(train2, aes(y=SalePrice, fill=ExterQual)) +geom\_boxplot(alpha=0.5)



#pairwise.wilcox.test(train2$SalePrice, train2$ExterQual)

## BsmtQual - Evaluates the height of the basement

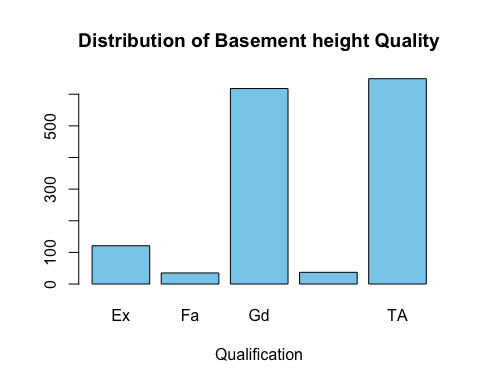
sum(is.na(train2$BsmtQual)) #0

## [1] 0

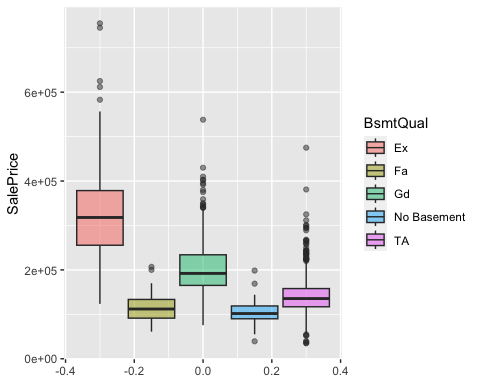
table(train2$BsmtQual)

##   
## Ex Fa Gd No Basement TA   
## 121 35 618 37 649

barplot(table(train2$BsmtQual), main = "Distribution of Basement height Quality",xlab = "Qualification",col = "skyblue")



#pie(table(train2$BsmtQual), main = "Distribution of BsmtQual", col = rainbow(length(levels(train2$BsmtQual))))  
  
# analysis with the target variable  
#ggplot(train2, aes(x=SalePrice, fill=BsmtQual)) +geom\_density(alpha=.5)  
ggplot(train2, aes(y=SalePrice, fill=BsmtQual)) + geom\_boxplot(alpha=0.5)



#pairwise.wilcox.test(train2$SalePrice, train2$BsmtQual)

## KitchenQual - Kitchen quality

sum(is.na(train2$KitchenQual)) #0

## [1] 0

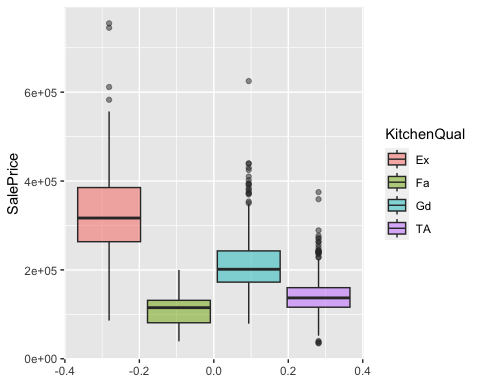
table(train2$KitchenQual)

##   
## Ex Fa Gd TA   
## 100 39 586 735

barplot(table(train2$KitchenQual), main = "Distribution of Kitchen Quality",xlab = "Qualification",col = "skyblue")



# analysis with the target variable  
#ggplot(train2, aes(x=SalePrice, fill=KitchenQual)) +geom\_density(alpha=.5)  
ggplot(train2, aes(y=SalePrice, fill=KitchenQual)) + geom\_boxplot(alpha=0.5)



#pairwise.wilcox.test(train2$SalePrice, train2$KitchenQual)  
#ks.test(train2$SalePrice[train2$KitchenQual=="TA"],train2$SalePrice[train2$KitchenQual=="Fa"]) # test if we can join certain categoris together to make a easier analysis ->rejected

## GarageFinish - Interior finish of the garage

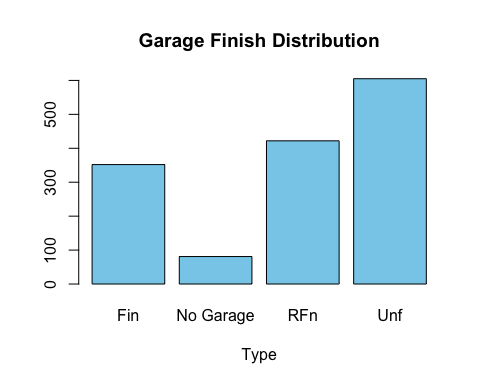
sum(is.na(train2$GarageFinish)) #0

## [1] 0

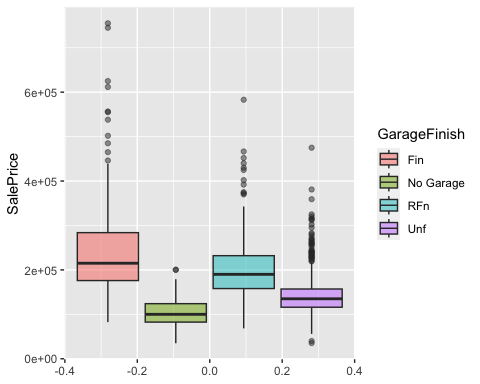
table(train2$GarageFinish)

##   
## Fin No Garage RFn Unf   
## 352 81 422 605

barplot(table(train2$GarageFinish), main = "Garage Finish Distribution",xlab = "Type",col = "skyblue")



# analysis with the target variable  
#ggplot(train2, aes(x=SalePrice, fill=GarageFinish)) +geom\_density(alpha=.5)  
ggplot(train2, aes(y=SalePrice, fill=GarageFinish)) +  
 geom\_boxplot(alpha=0.5)



#pairwise.wilcox.test(train2$SalePrice, train2$GarageFinish)

## FireplaceQu - Fireplace quality

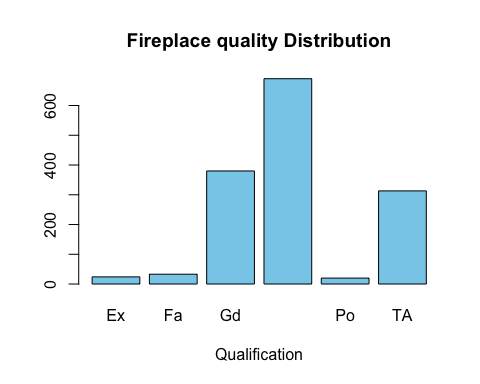
sum(is.na(train2$FireplaceQu)) #0

## [1] 0

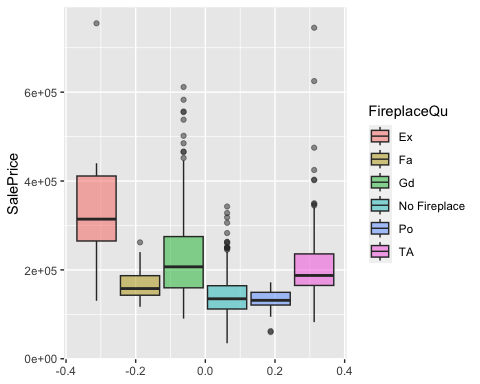
table(train2$FireplaceQu)

##   
## Ex Fa Gd No Fireplace Po TA   
## 24 33 380 690 20 313

barplot(table(train2$FireplaceQu), main = "Fireplace quality Distribution",xlab = "Qualification",col = "skyblue")



# analysis with the target variable  
#ggplot(train2, aes(x=SalePrice, fill=FireplaceQu)) +geom\_density(alpha=.5)  
ggplot(train2, aes(y=SalePrice, fill=FireplaceQu)) +  
 geom\_boxplot(alpha=0.5)



#pairwise.wilcox.test(train2$SalePrice, train2$FireplaceQu)

## Foundation - Type of foundation

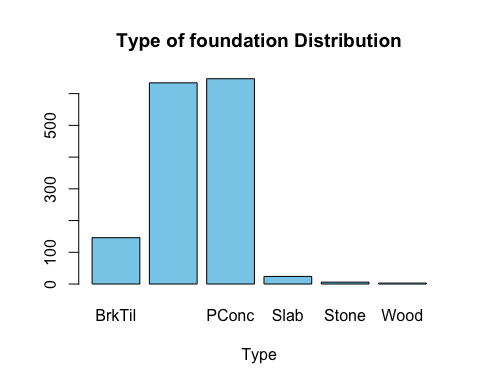
sum(is.na(train2$Foundation)) #0

## [1] 0

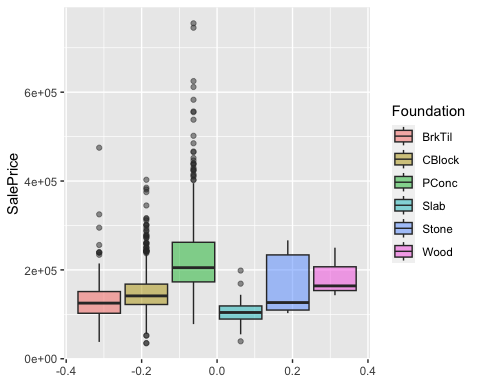
table(train2$Foundation)

##   
## BrkTil CBlock PConc Slab Stone Wood   
## 146 634 647 24 6 3

barplot(table(train2$Foundation), main = "Type of foundation Distribution",xlab = "Type",col = "skyblue")



# analysis with the target variable  
#ggplot(train2, aes(x=SalePrice, fill=Foundation)) +geom\_density(alpha=.5)  
ggplot(train2, aes(y=SalePrice, fill=Foundation)) +  
 geom\_boxplot(alpha=0.5)



#pairwise.wilcox.test(train2$SalePrice, train2$Foundation)

## GarageType - Garage location

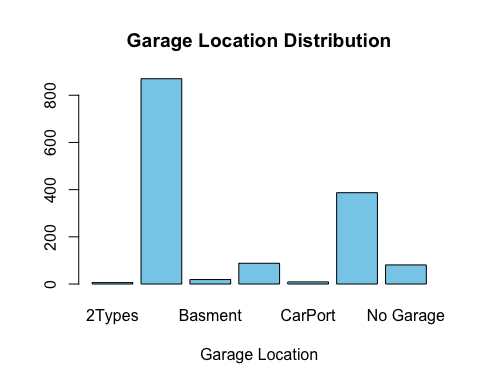
sum(is.na(train2$GarageType)) #0

## [1] 0

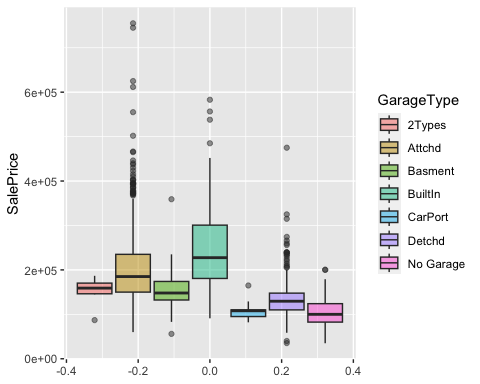
table(train2$GarageType)

##   
## 2Types Attchd Basment BuiltIn CarPort Detchd No Garage   
## 6 870 19 88 9 387 81

barplot(table(train2$GarageType), main = "Garage Location Distribution",xlab = "Garage Location",col = "skyblue")



# analysis with the target variable  
#ggplot(train2, aes(x=SalePrice, fill=GarageType)) +geom\_density(alpha=.5)  
ggplot(train2, aes(y=SalePrice, fill=GarageType)) +  
 geom\_boxplot(alpha=0.5)



#pairwise.wilcox.test(train2$SalePrice, train2$GarageType)

## MSSubClass

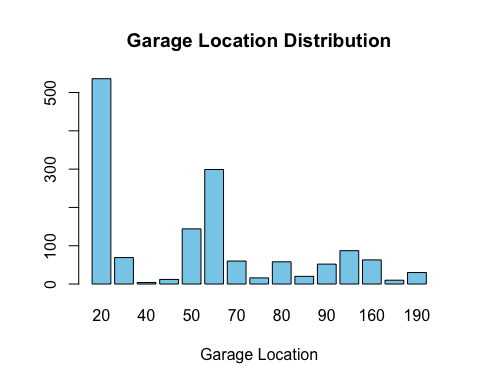
sum(is.na(train2$MSSubClass)) #0

## [1] 0

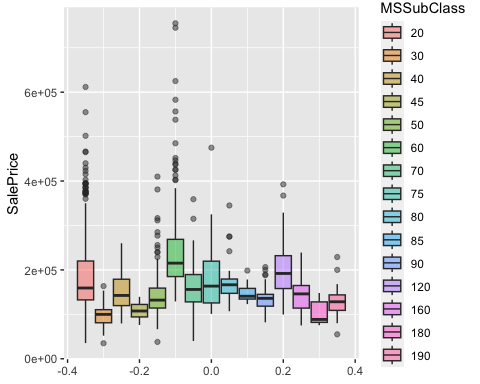
table(train2$MSSubClass) # +classes

##   
## 20 30 40 45 50 60 70 75 80 85 90 120 160 180 190   
## 536 69 4 12 144 299 60 16 58 20 52 87 63 10 30

barplot(table(train2$MSSubClass), main = "Garage Location Distribution",xlab = "Garage Location",col = "skyblue")



# analysis with the target variable  
#ggplot(train2, aes(x=SalePrice, fill=MSSubClass)) +geom\_density(alpha=.5)  
ggplot(train2, aes(y=SalePrice, fill=MSSubClass)) +  
 geom\_boxplot(alpha=0.5)



#pairwise.wilcox.test(train2$SalePrice, train2$MSSubClass)

## Missing values and imputation

First, we check which variables have NA’s and then we decide whether we impute or not.

#skim(train2)  
sum(is.na(train2$LotFrontage))

## [1] 259

sum(is.na(train2$MasVnrArea))

## [1] 10

sum(is.na(train2$GarageYrBlt))

## [1] 81

We exclude the imputation of the GarageYrBlt feature because it has no sense to impute this variable.

mice\_imp<-mice(train2[, !names(train2) %in% "GarageYrBlt"],method = "cart")

##   
## iter imp variable  
## 1 1 LotFrontage MasVnrArea  
## 1 2 LotFrontage MasVnrArea  
## 1 3 LotFrontage MasVnrArea  
## 1 4 LotFrontage MasVnrArea  
## 1 5 LotFrontage MasVnrArea  
## 2 1 LotFrontage MasVnrArea  
## 2 2 LotFrontage MasVnrArea  
## 2 3 LotFrontage MasVnrArea  
## 2 4 LotFrontage MasVnrArea  
## 2 5 LotFrontage MasVnrArea  
## 3 1 LotFrontage MasVnrArea  
## 3 2 LotFrontage MasVnrArea  
## 3 3 LotFrontage MasVnrArea  
## 3 4 LotFrontage MasVnrArea  
## 3 5 LotFrontage MasVnrArea  
## 4 1 LotFrontage MasVnrArea  
## 4 2 LotFrontage MasVnrArea  
## 4 3 LotFrontage MasVnrArea  
## 4 4 LotFrontage MasVnrArea  
## 4 5 LotFrontage MasVnrArea  
## 5 1 LotFrontage MasVnrArea  
## 5 2 LotFrontage MasVnrArea  
## 5 3 LotFrontage MasVnrArea  
## 5 4 LotFrontage MasVnrArea  
## 5 5 LotFrontage MasVnrArea

## Warning: Number of logged events: 50

imputed\_data<-complete(mice\_imp)

We proceed to validate the imputation

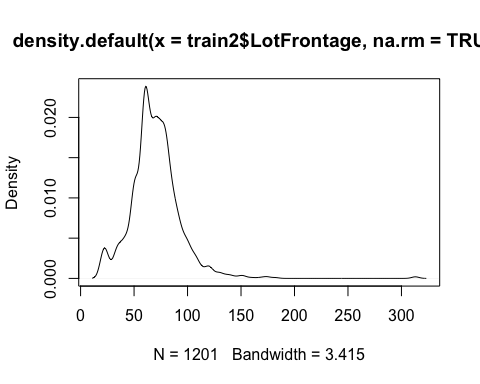
# LotFrontage validation  
summary(imputed\_data$LotFrontage)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 21.00 60.00 70.00 70.83 80.00 313.00

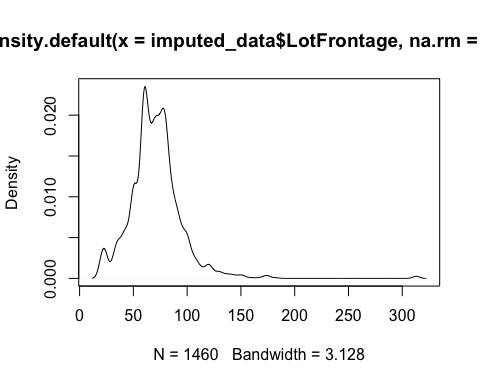
summary(train2$LotFrontage)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 21.00 59.00 69.00 70.05 80.00 313.00 259

plot(density(train2$LotFrontage,na.rm=TRUE))



plot(density(imputed\_data$LotFrontage,na.rm=TRUE))

 The imputation doesn’t change much the density nor summary.

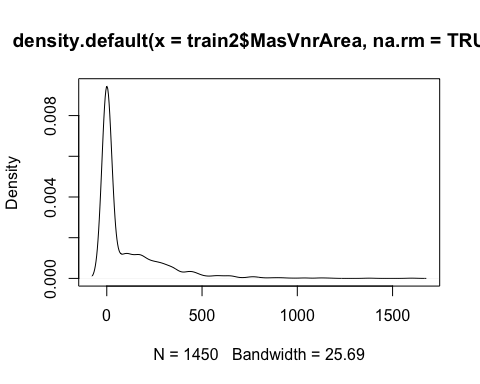
# MasVnrArea validation  
summary(imputed\_data$MasVnrArea)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0 0.0 0.0 103.7 166.0 1600.0

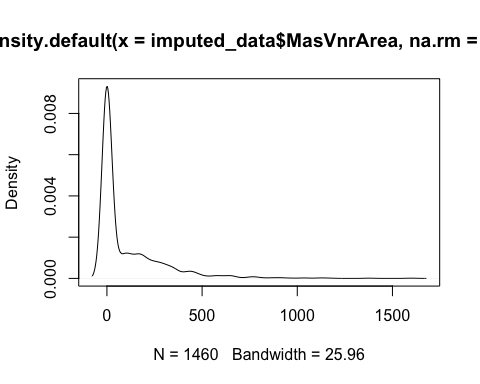
summary(train2$MasVnrArea)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.0 0.0 0.0 103.2 164.0 1600.0 10

plot(density(train2$MasVnrArea,na.rm=TRUE))



plot(density(imputed\_data$MasVnrArea,na.rm=TRUE))

 The imputation doesnt change much the density nor summary

## Data Quality Exploration

To check the quality of the dataset we have a count on each individual of the univariate outliers. We can see that there are 7 instances that have 4 or more univariate outliers, therefore are candidates to delete from the analysis.

max(imputed\_data$univ\_outl\_count)

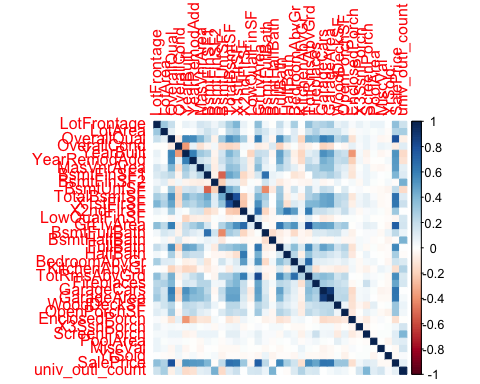
## [1] 10

ind\_delete<-imputed\_data[which(imputed\_data$univ\_outl\_count > 3),]  
imputed\_data<-imputed\_data[-c(ind\_delete)$Id,]

# Correlation Matrix

The correlation matrix shows a moderate negative correlation to the BsmtUnfSF (then the less Unfinished square feet of basement area, the more univ outliers) and a significant positive correlation to both BsmtFinSF2, Type 2 finished square feet, and ScreenPorch.

cols <- num\_cols[! num\_cols %in% c("Id","GarageYrBlt")]  
df\_of\_interest <- imputed\_data[,c(cols,"univ\_outl\_count")]  
cor\_outl = cor(df\_of\_interest)  
par(mfrow=c(1,1))  
corrplot(cor\_outl, method = 'color')



cor\_outl[34, ]

## LotFrontage LotArea OverallQual OverallCond YearBuilt   
## 0.35216627 0.26760706 0.79585433 -0.07670080 0.52598160   
## YearRemodAdd MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF   
## 0.50827902 0.49089068 0.40869598 -0.01287439 0.21580512   
## TotalBsmtSF X1stFlrSF X2ndFlrSF LowQualFinSF GrLivArea   
## 0.65090160 0.63120195 0.31955392 -0.03309214 0.73959382   
## BsmtFullBath BsmtHalfBath FullBath HalfBath BedroomAbvGr   
## 0.22750432 -0.01689242 0.56178277 0.28454057 0.16740870   
## KitchenAbvGr TotRmsAbvGrd Fireplaces GarageCars GarageArea   
## -0.13752026 0.54089962 0.46953187 0.64249623 0.63035535   
## WoodDeckSF OpenPorchSF EnclosedPorch X3SsnPorch ScreenPorch   
## 0.32676978 0.32454345 -0.13626825 0.04468370 0.10959267   
## PoolArea MiscVal YrSold SalePrice univ\_outl\_count   
## 0.09826675 -0.02317955 -0.02774883 1.00000000 0.08206935

colnames(cor\_outl)[which.min(cor\_outl[34, ])];min(cor\_outl[34, ])

## [1] "KitchenAbvGr"

## [1] -0.1375203

colnames(cor\_outl)[which.max(cor\_outl[34,1:33])];max(cor\_outl[34,1:33])

## [1] "OverallQual"

## [1] 0.7958543

# Multivariate outliers

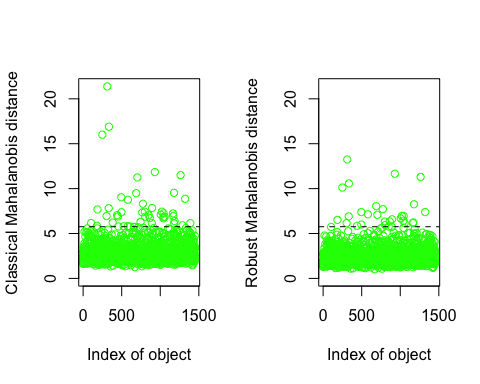
Filter variables in order to have only those with similar distribution of values. For that reason, we have filtered the values that had more unique values than 40. Then, we have excluded the variables that were centered in the ‘0’ value.

n\_cols <- names(imputed\_data)[sapply(imputed\_data, is.numeric)]  
n\_cols <- n\_cols[n\_cols != "Id"]  
df\_of\_interest <- imputed\_data[,c(n\_cols)]  
  
threshold <- 40  
df\_cols <- sapply(df\_of\_interest, function(x) length(unique(x)) > threshold)  
df\_of\_interest <- df\_of\_interest[, df\_cols]  
  
names(df\_of\_interest)

## [1] "LotFrontage" "LotArea" "YearBuilt" "YearRemodAdd"   
## [5] "MasVnrArea" "BsmtFinSF1" "BsmtFinSF2" "BsmtUnfSF"   
## [9] "TotalBsmtSF" "X1stFlrSF" "X2ndFlrSF" "GrLivArea"   
## [13] "GarageArea" "WoodDeckSF" "OpenPorchSF" "EnclosedPorch"  
## [17] "ScreenPorch" "SalePrice"

res.out = Moutlier(df\_of\_interest[, !(names(df\_of\_interest) %in% c("MasVnrArea","BsmtFinSF1", "BsmtFinSF2", "X2ndFlrSF", "WoodDeckSF", "EnclosedPorch", "ScreenPorch"))], quantile = 0.9995, col="green")

## Warning in covMcd(X): The covariance matrix has become singular during  
## the iterations of the MCD algorithm.  
## There are 588 observations (in the entire dataset of 1453 obs.) lying  
## on the hyperplane with equation a\_1\*(x\_i1 - m\_1) + ... + a\_p\*(x\_ip -  
## m\_p) = 0 with (m\_1, ..., m\_p) the mean of these observations and  
## coefficients a\_i from the vector a <- c(0, 0, -0.7071068, 0.7071068, 0,  
## 0, 0, 0, 0, 0, 0)



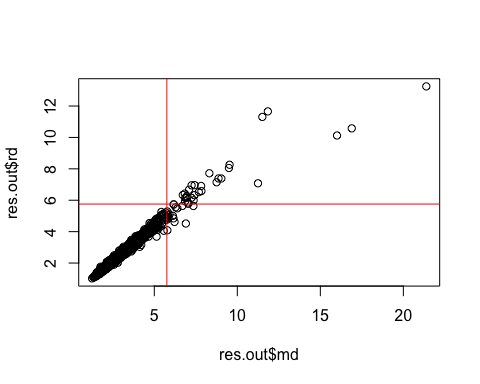
which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff))

## 186 250 305 314 333 336 441 496 497 584 692 707 779 804 808 855   
## 186 249 304 313 332 335 440 495 496 581 688 703 775 800 804 850   
## 869 899 935 1012 1025 1046 1049 1062 1170 1174 1183 1269 1329   
## 864 894 930 1007 1020 1041 1044 1057 1165 1169 1178 1264 1323

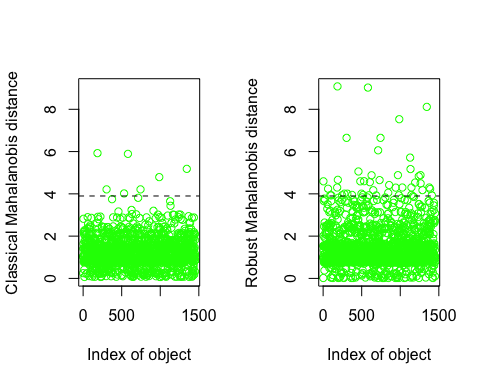
length(which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff))) #44

## [1] 29

par(mfrow=c(1,1))  
plot( res.out$md, res.out$rd )  
abline(h=res.out$cutoff, col="red")  
abline(v=res.out$cutoff, col="red")



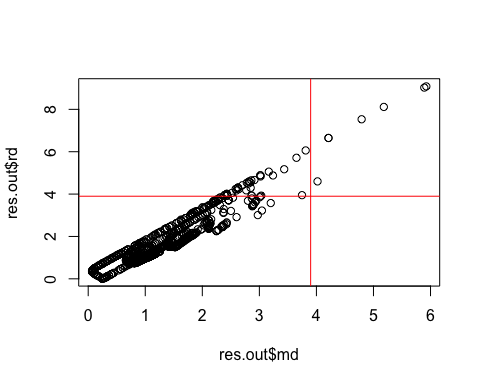
# take off m outliers  
#df = df[-which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff)),]  
  
# 2. Generally the houses built latetly are the ones with most quality  
res.out <- Moutlier(imputed\_data[,c("YearBuilt","OverallQual")], quantile = 0.9995, col="green")



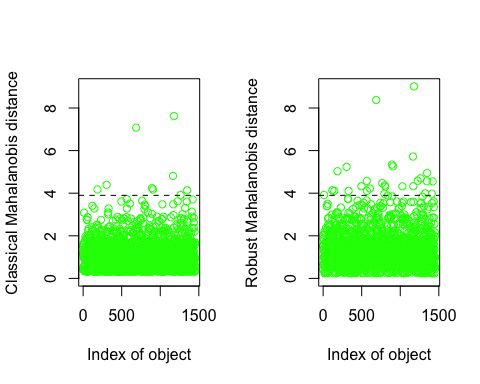
which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff))

## 186 305 534 584 748 992 1350   
## 186 304 531 581 744 987 1344

par(mfrow=c(1,1))  
plot( res.out$md, res.out$rd )  
abline(h=res.out$cutoff, col="red")  
abline(v=res.out$cutoff, col="red")



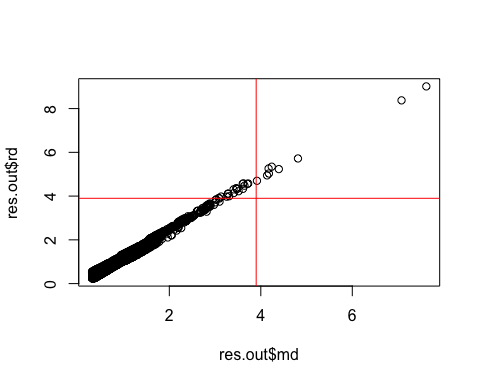
# 3. Generally the houses with more area are the ones with more rooms  
res.out <- Moutlier(imputed\_data[,c("GrLivArea","TotRmsAbvGrd")], quantile = 0.9995, col="green")



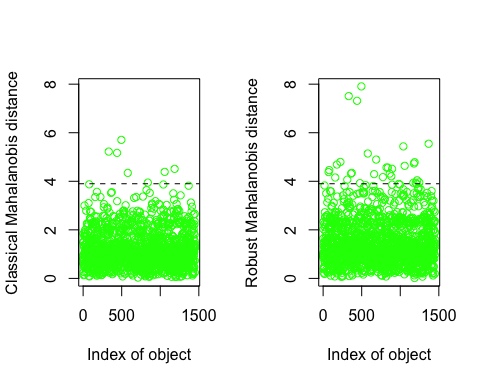
which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff))

## 186 305 692 898 911 1170 1183 1269 1354   
## 186 304 688 893 906 1165 1178 1264 1348

par(mfrow=c(1,1))  
plot( res.out$md, res.out$rd )  
abline(h=res.out$cutoff, col="red")  
abline(v=res.out$cutoff, col="red")



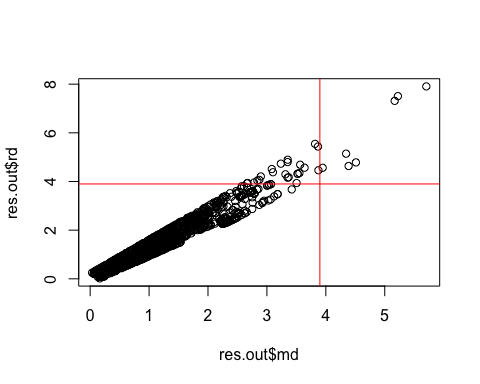
# 4. Generally the houses with more total area are the ones with more garage area also  
res.out <- Moutlier(imputed\_data[,c("GarageArea","TotalBsmtSF")], quantile = 0.9995, col="green")



which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff))

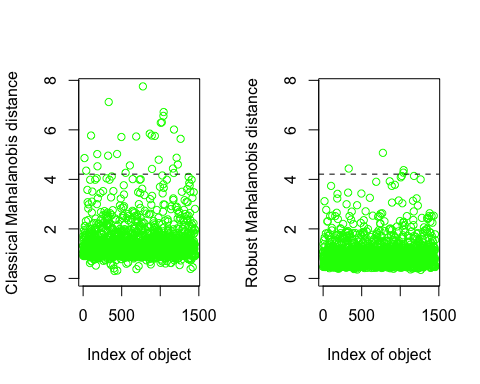
## 333 441 497 582 844 1062 1191   
## 332 440 496 579 839 1057 1186

par(mfrow=c(1,1))  
plot( res.out$md, res.out$rd )  
abline(h=res.out$cutoff, col="red")  
abline(v=res.out$cutoff, col="red")



# 5. Moutliers about total areas zones  
res.out <- Moutlier(imputed\_data[,c("X1stFlrSF","TotalBsmtSF","GrLivArea")], quantile = 0.9995, col="green") #GarageArea, si se lo sumo hay menos outliers

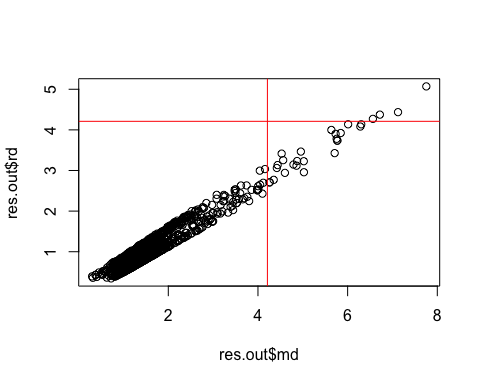
## Warning in covMcd(X): The covariance matrix has become singular during  
## the iterations of the MCD algorithm.  
## There are 657 observations (in the entire dataset of 1453 obs.) lying  
## on the plane with equation 0.70711 (x\_i1-m\_1) + 2.7219e-15 (x\_i2-m\_2) +  
## -0.70711 (x\_i3-m\_3) = 0 with (m\_1,m\_2) the mean of these observations.



which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff))

## 333 779 1046 1049   
## 332 775 1041 1044

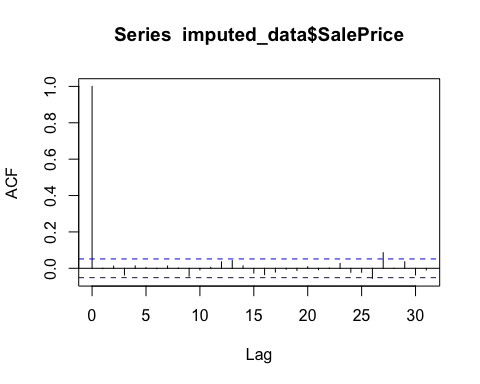
par(mfrow=c(1,1))  
plot( res.out$md, res.out$rd )  
abline(h=res.out$cutoff, col="red")  
abline(v=res.out$cutoff, col="red")



# Profiling

To test for autocorrelation the acf() function is used, of which the result can be seen below: target is not normally distributed and autocorrelation is found, so a randomized dataframe is reset

acf(imputed\_data$SalePrice)



#duplicate of the section target exploration  
shapiro.test(imputed\_data$SalePrice) ## Normality test of the target variable ->>> # We test the null hypothesis H0: variable is normal, H1: variable not normal. # The p-value is less than 0.05, therefore we reject H0. # For that we will not consider the target variable as normally distributed

##   
## Shapiro-Wilk normality test  
##   
## data: imputed\_data$SalePrice  
## W = 0.86925, p-value < 2.2e-16

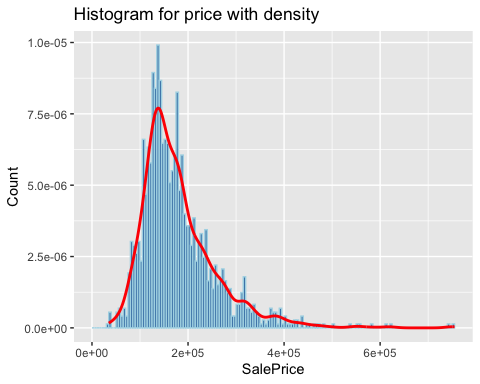
ks.test(imputed\_data$SalePrice, 'pnorm', mean(imputed\_data$SalePrice), sd(imputed\_data$SalePrice))

## Warning in ks.test.default(imputed\_data$SalePrice, "pnorm",  
## mean(imputed\_data$SalePrice), : ties should not be present for the  
## Kolmogorov-Smirnov test

##   
## Asymptotic one-sample Kolmogorov-Smirnov test  
##   
## data: imputed\_data$SalePrice  
## D = 0.124, p-value < 2.2e-16  
## alternative hypothesis: two-sided

#qqnorm(imputed\_data$SalePrice) + qqline(imputed\_data$SalePrice)  
#hist(imputed\_data$SalePrice, prob = TRUE, col = 'lightblue', main = 'SalePrice Distribution', xlab = 'SalePrice') + lines(density(imputed\_data$SalePrice), col = 'red', lwd = 2)  
  
ggplot(data=imputed\_data, aes(SalePrice, y = after\_stat(density))) +  
 geom\_histogram(breaks=seq(0, max(imputed\_data$SalePrice), by=5000),  
 col='lightblue',  
 fill='steelblue') +  
 geom\_density(lwd=1,  
 col='red') +  
 labs(title="Histogram for price with density", x="SalePrice", y="Count")

## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.  
## ℹ Please use `linewidth` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.



# Variables with most association with the response variable

For the quantitative variables there are some variable that are highly significant positively correlated (r > 0.50, p ~ 0) to the target variable, SalesPrice: OverallQual, GrLivArea, GarageCars, GarageArea, TotalBsmtSF, X1stFlrSF, FullBath, TotRmsAbvGrd, YearBuilt and YearRemodAdd. This seems logical as the higher the overall condition rating of the house is and the bigger the area above grade living, the more it would cost. There isn’t any variable which is highly significant negatively correlated to the SalePrice (r < -0.50 p = 0).

For the qualitative variables it is clear that the Neighborhood explains the most variance in the SalePrice variable (R^2 = 0.546, p ~ 0).  
The influence of the other qualitative variables (with R^2 > 0.3) are in order: Neighborhood, ExterQual, BsmtQual, KitchenQual, f.GarageArea, GarageFinish. EnclosedPorch\_binary, Secondfloor and Pool\_binary are poorly associated as they have a R^2-value under 4%.

res.con <- condes(imputed\_data,35) #names(imputed\_data)[35] - [1] "SalePrice"  
res.con$quanti

## correlation p.value  
## OverallQual 0.79585433 1.800118e-318  
## GrLivArea 0.73959382 8.947769e-252  
## TotalBsmtSF 0.65090160 7.455967e-176  
## GarageCars 0.64249623 5.837898e-170  
## X1stFlrSF 0.63120195 2.511377e-162  
## GarageArea 0.63035535 9.107151e-162  
## FullBath 0.56178277 1.225399e-121  
## TotRmsAbvGrd 0.54089962 3.400072e-111  
## YearBuilt 0.52598160 3.578620e-104  
## YearRemodAdd 0.50827902 2.782822e-96  
## MasVnrArea 0.49089068 5.694482e-89  
## Fireplaces 0.46953187 1.503171e-80  
## BsmtFinSF1 0.40869598 1.329980e-59  
## LotFrontage 0.35216627 1.125496e-43  
## WoodDeckSF 0.32676978 1.679594e-37  
## OpenPorchSF 0.32454345 5.485499e-37  
## X2ndFlrSF 0.31955392 7.509846e-36  
## HalfBath 0.28454057 1.826130e-28  
## LotArea 0.26760706 2.999057e-25  
## BsmtFullBath 0.22750432 1.635036e-18  
## BsmtUnfSF 0.21580512 9.004815e-17  
## BedroomAbvGr 0.16740870 1.352061e-10  
## ScreenPorch 0.10959267 2.832892e-05  
## PoolArea 0.09826675 1.756995e-04  
## univ\_outl\_count 0.08206935 1.742542e-03  
## OverallCond -0.07670080 3.439104e-03  
## EnclosedPorch -0.13626825 1.846376e-07  
## KitchenAbvGr -0.13752026 1.420791e-07

#done before for selecting  
res.con$quali

## R2 p.value  
## Neighborhood 0.54724351 1.747377e-225  
## ExterQual 0.48825569 3.437545e-210  
## BsmtQual 0.47146120 1.101824e-198  
## KitchenQual 0.46279482 6.343171e-195  
## GarageFinish 0.31012463 2.620955e-116  
## FireplaceQu 0.29573420 1.631423e-107  
## Foundation 0.25767350 4.584555e-91  
## GarageType 0.25210757 1.026767e-87  
## MSSubClass 0.24792113 5.081782e-79  
## BsmtFinType1 0.21223984 1.479192e-71  
## OpenPorch\_binary 0.17109800 3.776037e-61  
## HasPorch\_binary 0.09173386 3.314957e-32  
## EnclosedPorch\_binary 0.03423831 1.171882e-12  
## secondfloor 0.01860862 1.791339e-07  
## Pool\_binary 0.01146405 4.321554e-05

## Define a polytomic factor ExterQual, evaluates the quality of the material on the exterior, and argue if the average price depends on the level of ExterQual

Define a polytomic factor f.age for the covariate car age according to its quartiles and argue if the average price depends on the level of age. Statistically justify the answer

Given the evidence below, we argue that the price is dependent on the level of ExterQual. Firstly, in the boxplots it can be observed that the better the quality (Q1) the lower the average of the house sale price. This is also seen clearly in the distribution plot below. —Secondly, the wilcoxon test shows that the means of these levels are not equal meaning that some relation exists between the factor and the response variable

We perform pairwise wilcoxon test to check for similar means. Looking at the boxplot and distribution plot we can already expect that these are not going to be similar. The result of the wilcoxon test indicates our hypothesis was right that there is a clear difference between the means of the different quartiles

pairwise.wilcox.test(imputed\_data$SalePrice, imputed\_data$ExterQual)

##   
## Pairwise comparisons using Wilcoxon rank sum test with continuity correction   
##   
## data: imputed\_data$SalePrice and imputed\_data$ExterQual   
##   
## Ex Fa Gd   
## Fa 7.1e-08 - -   
## Gd < 2e-16 3.0e-09 -   
## TA < 2e-16 9.9e-08 < 2e-16  
##   
## P value adjustment method: holm

## LowAge LowMidAge HighMidAge  
## LowMidAge <2e-16 - -   
## HighMidAge <2e-16 <2e-16 -   
## HighAge <2e-16 <2e-16 <2e-16  
  
#We perform pairwise wilcoxon test to check for similar means. Looking at the boxplot   
#and distribution plot we can already expect that these are not going to be similar. The   
#result of the wilcoxon test indicates our hypothesis was right that there is a clear   
#difference between the means of the different quartiles

## Modeling

To create a model, first we split the train data (imputed\_data) into train\_new and test\_new. We need the test\_new to validate the accuracy of the model with data that has the target variable and the model has not used. To check if the model has overfitting.

set.seed(123)  
rows <- sample(nrow(imputed\_data), .7 \* nrow(imputed\_data))  
train\_new <- imputed\_data[rows, ]  
test\_new <- imputed\_data[-rows, ]

Before building the model, we should first study the multicolinearity and check if there are any predictors that are highly correlated.

num\_cols <- names(train\_new)[sapply(train\_new, is.numeric)]  
num\_data<-train\_new[,c(num\_cols)]  
descrCor <- cor(num\_data)  
highCorr <- sum(abs(descrCor[upper.tri(descrCor)]) > .99)   
  
max\_cor\_value <-max(abs(descrCor[upper.tri(descrCor)])) # maximum correlated  
which(descrCor == max\_cor\_value, arr.ind = TRUE) # garageArea and garagecars with 88% correlation

## row col  
## GarageArea 26 25  
## GarageCars 25 26

vec <- as.vector(descrCor)  
#sort(unique(vec), decreasing = TRUE)  
sort(unique(vec), decreasing = TRUE)[3];which(descrCor == sort(unique(vec), decreasing = TRUE)[3], arr.ind = TRUE)

## [1] 0.8341127

## row col  
## TotRmsAbvGrd 23 16  
## GrLivArea 16 23

sort(unique(vec), decreasing = TRUE)[4];which(descrCor == sort(unique(vec), decreasing = TRUE)[4], arr.ind = TRUE)

## [1] 0.8000323

## row col  
## SalePrice 35 4  
## OverallQual 4 35

sort(unique(vec), decreasing = TRUE)[5];which(descrCor == sort(unique(vec), decreasing = TRUE)[5], arr.ind = TRUE)

## [1] 0.7926493

## row col  
## X1stFlrSF 13 12  
## TotalBsmtSF 12 13

We have seen that the previous variables are very correlated, GarageArea with GarageCars with 88% correlation, TotRmsAbvGrd and GrLivArea with 83% correlation, X1stFlrSF and TotalBsmtSF with 81% correlation and SalePrice with OverallQual with 79% what is indeed our target variable. We will not, for now, in the modeling process consider excluding some of the variables that are very correlated.

We start by building the model with a manual stepwise technique, we will check the R^2, RSS and AIC metrics to select the better model. We can start by adding the more correlated numerical variables to the target variable.

display(m0 <- lm(SalePrice ~ 1, data = train\_new))  
display(m0 <- lm(log(SalePrice) ~ 1, data = train\_new))  
display(m1 <- lm(log(SalePrice) ~ OverallQual , data = train\_new))  
#plot(m1)  
anova(m0,m1) # rss  
as.matrix(AIC(m0, m1))   
  
display(m2 <- lm(log(SalePrice) ~ OverallQual\*log(GrLivArea), data = train\_new)) # with interaction  
display(m2 <- lm(log(SalePrice) ~ OverallQual+log(GrLivArea), data = train\_new)) # without interaction  
#plot(m2)  
anova(m1,m2)  
as.matrix(AIC(m1, m2))  
  
display(m3 <- lm(log(SalePrice) ~OverallQual+log(GrLivArea)\*GarageCars , data = train\_new))  
display(m3 <- lm(log(SalePrice) ~OverallQual+log(GrLivArea)+GarageCars , data = train\_new))  
anova(m2,m3)   
as.matrix(AIC(m2, m3))   
summary(m3)  
  
display(m4 <- lm(log(SalePrice)~OverallQual+log(GrLivArea)+GarageCars+GarageArea , data = train\_new))  
anova(m3,m4)  
as.matrix(AIC(m3, m4))  
summary(m4)  
  
display(m5<- lm(log(SalePrice) ~OverallQual +log(GrLivArea)+GarageCars+GarageArea+TotalBsmtSF , data = train\_new))  
anova(m4,m5)  
summary(m5)   
as.matrix(AIC(m4, m5))   
  
display(m6<- lm(log(SalePrice) ~OverallQual +log(GrLivArea)+GarageCars+GarageArea+TotalBsmtSF+ X1stFlrSF, data = train\_new))  
anova(m5,m6)  
summary(m6)   
as.matrix(AIC(m5, m6)) #does not improve  
  
display(m7<- lm(log(SalePrice) ~OverallQual +log(GrLivArea)+GarageCars+TotalBsmtSF+FullBath, data = train\_new))  
anova(m5,m7) # does not improve much  
summary(m7)   
as.matrix(AIC(m5, m7))   
  
display(m8<- lm(log(SalePrice) ~OverallQual+log(GrLivArea)+GarageCars+GarageArea+TotalBsmtSF+TotRmsAbvGrd, data = train\_new))  
anova(m5,m8) # not improve  
as.matrix(AIC(m7, m8))   
  
display(m9<- lm(log(SalePrice) ~OverallQual+log(GrLivArea)+GarageCars+GarageArea+TotalBsmtSF+YearBuilt, data = train\_new))  
anova(m5,m9)   
summary(m8)  
as.matrix(AIC(m8, m9)) # improves  
  
display(m10<- lm(log(SalePrice) ~OverallQual+log(GrLivArea)+GarageCars+GarageArea+TotalBsmtSF+YearBuilt+YearRemodAdd, data = train\_new))  
anova(m9,m10)   
summary(m10)  
as.matrix(AIC(m9, m10))   
  
display(m11<- lm(log(SalePrice) ~OverallQual+log(GrLivArea)+GarageCars+GarageArea+TotalBsmtSF+YearBuilt+YearRemodAdd+MasVnrArea, data = train\_new))  
anova(m10,m11)   
summary(m11)  
as.matrix(AIC(m10, m11)) # not much better  
  
  
display(m12<- lm(log(SalePrice) ~OverallQual+log(GrLivArea)+GarageCars+GarageArea+TotalBsmtSF+YearBuilt+YearRemodAdd+Fireplaces, data = train\_new))  
anova(m10,m12)   
summary(m12) # r^2 better  
as.matrix(AIC(m10, m12))   
  
  
display(m13<- lm(log(SalePrice) ~OverallQual+log(GrLivArea)+GarageCars+GarageArea+TotalBsmtSF+YearBuilt+YearRemodAdd+Fireplaces+BsmtFinSF1, data = train\_new))  
anova(m12,m13)   
summary(m13) # r^2 NOT better , rss and AIC yes -> keep  
as.matrix(AIC(m10, m13))  
  
display(m14<- lm(log(SalePrice) ~OverallQual+log(GrLivArea)+GarageCars+GarageArea+TotalBsmtSF+YearBuilt+YearRemodAdd+Fireplaces+BsmtFinSF1+LotFrontage, data = train\_new))  
anova(m13,m14) # R^2 bbetter -> keep  
summary(m14)  
as.matrix(AIC(m13, m14))  
  
display(m15<- lm(log(SalePrice) ~OverallQual+log(GrLivArea)+GarageCars+GarageArea+TotalBsmtSF+YearBuilt+YearRemodAdd+Fireplaces+BsmtFinSF1+LotFrontage+WoodDeckSF, data = train\_new))  
anova(m14,m15)   
summary(m15)   
as.matrix(AIC(m14, m15))  
  
# delete garagearea bv very correlated to garage  
display(m15<- lm(log(SalePrice) ~OverallQual+log(GrLivArea)+GarageCars+TotalBsmtSF+YearBuilt+YearRemodAdd+Fireplaces+BsmtFinSF1+LotFrontage+WoodDeckSF, data = train\_new))  
anova(m14,m15)   
summary(m15)   
as.matrix(AIC(m14, m15))

We can also use the automatic stepwise() and see which are the variables that will create a model with higher accuracy.

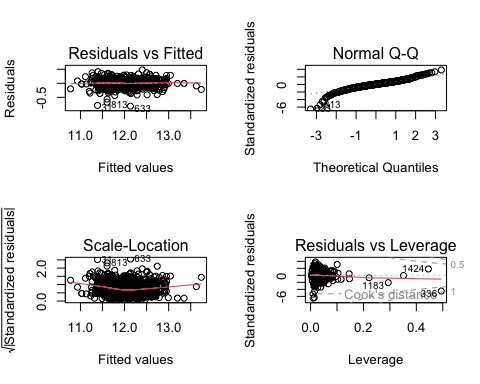
mod.fow <- stats::step(lm(SalePrice~. , data = train\_new), trace = FALSE,  
 direction = "forward")  
summary(mod.fow)

##   
## Call:  
## lm(formula = SalePrice ~ Id + LotFrontage + LotArea + OverallQual +   
## OverallCond + YearBuilt + YearRemodAdd + MasVnrArea + BsmtFinSF1 +   
## BsmtFinSF2 + BsmtUnfSF + TotalBsmtSF + X1stFlrSF + X2ndFlrSF +   
## LowQualFinSF + GrLivArea + BsmtFullBath + BsmtHalfBath +   
## FullBath + HalfBath + BedroomAbvGr + KitchenAbvGr + TotRmsAbvGrd +   
## Fireplaces + GarageCars + GarageArea + WoodDeckSF + OpenPorchSF +   
## EnclosedPorch + X3SsnPorch + ScreenPorch + PoolArea + MiscVal +   
## YrSold + Neighborhood + ExterQual + BsmtQual + KitchenQual +   
## GarageFinish + FireplaceQu + Foundation + GarageType + MSSubClass +   
## BsmtFinType1 + univ\_outl\_count + secondfloor + OpenPorch\_binary +   
## EnclosedPorch\_binary + HasPorch\_binary + Pool\_binary, data = train\_new)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -94482 -10487 836 10598 155602   
##   
## Coefficients: (4 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.271e+06 1.219e+06 1.042 0.297554   
## Id 2.175e+00 1.873e+00 1.162 0.245666   
## LotFrontage 1.093e+02 4.720e+01 2.315 0.020812 \*   
## LotArea 7.958e-01 1.384e-01 5.750 1.22e-08 \*\*\*  
## OverallQual 8.269e+03 1.175e+03 7.038 3.85e-12 \*\*\*  
## OverallCond 5.704e+03 9.499e+02 6.005 2.77e-09 \*\*\*  
## YearBuilt 4.411e+02 9.510e+01 4.638 4.03e-06 \*\*\*  
## YearRemodAdd 1.107e+02 6.161e+01 1.797 0.072716 .   
## MasVnrArea 2.238e+01 5.718e+00 3.914 9.77e-05 \*\*\*  
## BsmtFinSF1 4.959e+01 5.997e+00 8.269 4.77e-16 \*\*\*  
## BsmtFinSF2 3.390e+01 8.434e+00 4.020 6.31e-05 \*\*\*  
## BsmtUnfSF 2.770e+01 5.399e+00 5.131 3.53e-07 \*\*\*  
## TotalBsmtSF NA NA NA NA   
## X1stFlrSF 4.874e+01 6.106e+00 7.981 4.36e-15 \*\*\*  
## X2ndFlrSF 7.976e+01 7.199e+00 11.080 < 2e-16 \*\*\*  
## LowQualFinSF 4.015e+01 2.149e+01 1.868 0.062079 .   
## GrLivArea NA NA NA NA   
## BsmtFullBath 1.072e+03 2.330e+03 0.460 0.645554   
## BsmtHalfBath 2.321e+03 4.583e+03 0.506 0.612669   
## FullBath 6.073e+03 2.688e+03 2.259 0.024103 \*   
## HalfBath 3.216e+03 2.445e+03 1.315 0.188690   
## BedroomAbvGr -8.040e+03 1.638e+03 -4.908 1.09e-06 \*\*\*  
## KitchenAbvGr -9.348e+03 7.078e+03 -1.321 0.186909   
## TotRmsAbvGrd 1.236e+03 1.141e+03 1.083 0.279000   
## Fireplaces 6.271e+03 3.014e+03 2.081 0.037718 \*   
## GarageCars 7.713e+03 2.606e+03 2.960 0.003161 \*\*   
## GarageArea -3.108e+00 9.040e+00 -0.344 0.731047   
## WoodDeckSF 1.188e+01 6.955e+00 1.708 0.088045 .   
## OpenPorchSF 2.888e+01 1.641e+01 1.760 0.078801 .   
## EnclosedPorch -7.118e+00 3.268e+01 -0.218 0.827603   
## X3SsnPorch 2.294e+01 2.445e+01 0.939 0.348224   
## ScreenPorch 3.314e+01 1.878e+01 1.764 0.078034 .   
## PoolArea -5.830e+02 2.025e+02 -2.879 0.004079 \*\*   
## MiscVal -2.865e+00 2.862e+00 -1.001 0.317108   
## YrSold -1.165e+03 5.993e+02 -1.944 0.052262 .   
## NeighborhoodBlueste -1.558e+03 2.641e+04 -0.059 0.952957   
## NeighborhoodBrDale -1.468e+04 1.254e+04 -1.171 0.241915   
## NeighborhoodBrkSide -1.596e+04 1.067e+04 -1.496 0.134986   
## NeighborhoodClearCr -2.873e+04 1.103e+04 -2.605 0.009344 \*\*   
## NeighborhoodCollgCr -2.550e+04 9.157e+03 -2.784 0.005474 \*\*   
## NeighborhoodCrawfor -1.457e+02 1.034e+04 -0.014 0.988757   
## NeighborhoodEdwards -2.796e+04 9.939e+03 -2.813 0.005012 \*\*   
## NeighborhoodGilbert -2.708e+04 9.501e+03 -2.850 0.004472 \*\*   
## NeighborhoodIDOTRR -3.126e+04 1.131e+04 -2.764 0.005821 \*\*   
## NeighborhoodMeadowV -2.229e+03 1.339e+04 -0.167 0.867773   
## NeighborhoodMitchel -3.433e+04 1.004e+04 -3.420 0.000655 \*\*\*  
## NeighborhoodNAmes -2.583e+04 9.630e+03 -2.682 0.007451 \*\*   
## NeighborhoodNoRidge -1.132e+03 1.055e+04 -0.107 0.914561   
## NeighborhoodNPkVill 2.180e+03 1.316e+04 0.166 0.868471   
## NeighborhoodNridgHt -9.179e+03 9.485e+03 -0.968 0.333413   
## NeighborhoodNWAmes -3.810e+04 9.847e+03 -3.869 0.000117 \*\*\*  
## NeighborhoodOldTown -2.988e+04 1.029e+04 -2.904 0.003772 \*\*   
## NeighborhoodSawyer -2.705e+04 9.994e+03 -2.706 0.006928 \*\*   
## NeighborhoodSawyerW -2.636e+04 9.583e+03 -2.750 0.006071 \*\*   
## NeighborhoodSomerst -1.243e+04 9.513e+03 -1.307 0.191637   
## NeighborhoodStoneBr 2.046e+04 1.026e+04 1.995 0.046377 \*   
## NeighborhoodSWISU -2.527e+04 1.178e+04 -2.146 0.032160 \*   
## NeighborhoodTimber -2.929e+04 9.994e+03 -2.931 0.003468 \*\*   
## NeighborhoodVeenker -5.733e+03 1.249e+04 -0.459 0.646238   
## ExterQualFa -2.408e+04 1.070e+04 -2.251 0.024642 \*   
## ExterQualGd -1.933e+04 5.322e+03 -3.632 0.000297 \*\*\*  
## ExterQualTA -2.119e+04 5.960e+03 -3.556 0.000396 \*\*\*  
## BsmtQualFa -2.367e+04 7.224e+03 -3.277 0.001090 \*\*   
## BsmtQualGd -2.513e+04 4.176e+03 -6.017 2.57e-09 \*\*\*  
## BsmtQualNo Basement 6.808e+03 1.211e+04 0.562 0.574230   
## BsmtQualTA -2.346e+04 5.025e+03 -4.670 3.47e-06 \*\*\*  
## KitchenQualFa -1.726e+04 7.204e+03 -2.396 0.016784 \*   
## KitchenQualGd -1.637e+04 4.115e+03 -3.979 7.48e-05 \*\*\*  
## KitchenQualTA -1.734e+04 4.596e+03 -3.772 0.000172 \*\*\*  
## GarageFinishNo Garage 3.268e+04 1.313e+04 2.488 0.013035 \*   
## GarageFinishRFn -8.615e+02 2.465e+03 -0.350 0.726776   
## GarageFinishUnf 2.261e+03 2.980e+03 0.759 0.448223   
## FireplaceQuFa -1.815e+04 8.444e+03 -2.149 0.031888 \*   
## FireplaceQuGd -7.204e+03 6.403e+03 -1.125 0.260861   
## FireplaceQuNo Fireplace -4.645e+01 7.748e+03 -0.006 0.995218   
## FireplaceQuPo -3.616e+02 9.388e+03 -0.039 0.969284   
## FireplaceQuTA -9.858e+03 6.706e+03 -1.470 0.141905   
## FoundationCBlock 2.621e+03 3.647e+03 0.719 0.472576   
## FoundationPConc 6.231e+03 3.972e+03 1.569 0.117060   
## FoundationSlab 2.138e+03 1.133e+04 0.189 0.850427   
## FoundationStone 1.218e+04 1.493e+04 0.816 0.414551   
## FoundationWood -3.021e+04 1.557e+04 -1.941 0.052559 .   
## GarageTypeAttchd 1.583e+04 1.175e+04 1.348 0.178091   
## GarageTypeBasment 1.773e+04 1.439e+04 1.232 0.218110   
## GarageTypeBuiltIn 1.807e+04 1.245e+04 1.451 0.147068   
## GarageTypeCarPort 1.298e+04 1.540e+04 0.843 0.399564   
## GarageTypeDetchd 2.006e+04 1.167e+04 1.719 0.086023 .   
## GarageTypeNo Garage NA NA NA NA   
## MSSubClass30 3.792e+03 5.489e+03 0.691 0.489816   
## MSSubClass40 -1.752e+03 1.830e+04 -0.096 0.923736   
## MSSubClass45 9.985e+01 1.009e+04 0.010 0.992104   
## MSSubClass50 9.941e+03 6.773e+03 1.468 0.142537   
## MSSubClass60 -8.147e+02 6.136e+03 -0.133 0.894395   
## MSSubClass70 1.491e+03 7.739e+03 0.193 0.847211   
## MSSubClass75 7.685e+03 1.212e+04 0.634 0.526327   
## MSSubClass80 6.880e+03 4.940e+03 1.393 0.164048   
## MSSubClass85 -7.789e+02 6.977e+03 -0.112 0.911133   
## MSSubClass90 -2.207e+04 7.329e+03 -3.011 0.002672 \*\*   
## MSSubClass120 -2.205e+04 4.765e+03 -4.626 4.27e-06 \*\*\*  
## MSSubClass160 -2.445e+04 8.469e+03 -2.887 0.003981 \*\*   
## MSSubClass180 -3.597e+04 1.509e+04 -2.384 0.017349 \*   
## MSSubClass190 -5.078e+03 8.035e+03 -0.632 0.527520   
## BsmtFinType1BLQ 4.590e+03 3.290e+03 1.395 0.163368   
## BsmtFinType1GLQ 4.251e+03 2.984e+03 1.425 0.154636   
## BsmtFinType1LwQ 5.497e+02 4.444e+03 0.124 0.901587   
## BsmtFinType1No Basement NA NA NA NA   
## BsmtFinType1Rec -7.877e+02 3.430e+03 -0.230 0.818443   
## BsmtFinType1Unf 4.424e+03 3.274e+03 1.351 0.177032   
## univ\_outl\_count -1.918e+01 2.544e+03 -0.008 0.993986   
## secondfloor1 -2.214e+04 7.084e+03 -3.126 0.001830 \*\*   
## OpenPorch\_binary1 -3.059e+03 5.028e+03 -0.608 0.543082   
## EnclosedPorch\_binary1 4.034e+03 6.934e+03 0.582 0.560872   
## HasPorch\_binary1 4.903e+02 5.199e+03 0.094 0.924886   
## Pool\_binary1 4.675e+05 1.328e+05 3.520 0.000453 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 23480 on 907 degrees of freedom  
## Multiple R-squared: 0.9255, Adjusted R-squared: 0.9165   
## F-statistic: 103.3 on 109 and 907 DF, p-value: < 2.2e-16

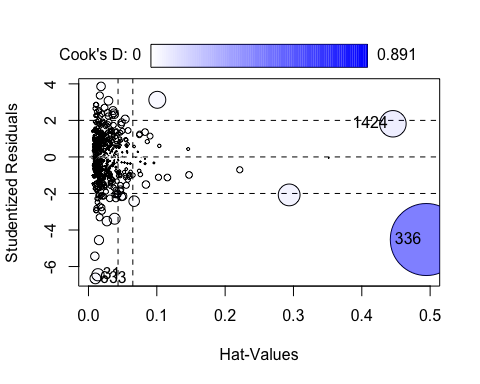
# Models:   
  
n00<- lm(log(SalePrice) ~ LotArea + OverallQual + OverallCond + YearBuilt +  
 MasVnrArea + BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF +  
 X1stFlrSF + X2ndFlrSF +  
 BsmtFullBath + FullBath + HalfBath + BedroomAbvGr +  
 KitchenAbvGr + Fireplaces + GarageCars +  
 OpenPorchSF + X3SsnPorch + ScreenPorch +  
 PoolArea, data = train\_new)  
#summary(n00)  
summary(n00)$r.squared

## [1] 0.904351

par(mfrow=c(2,2))  
plot(n00)



par(mfrow=c(1,1))  
influencePlot(n00)



## StudRes Hat CookD  
## 31 -6.449646 0.0138739 0.02555920  
## 336 -4.521383 0.4944163 0.89128177  
## 1424 1.814848 0.4455447 0.12002800  
## 633 -6.649463 0.0100953 0.01964319

#corrplot(descrCor, method = 'number')   
# check if there are any correlated predictors  
treshold\_vif<-function(rsq){  
 vif<- 1/(1-rsq)  
 return(vif)  
}  
  
treshold\_vif(summary(n00)$r.squared)

## [1] 10.4549

vif(n00) # there are not

## LotArea OverallQual OverallCond YearBuilt MasVnrArea BsmtFinSF1   
## 1.214912 3.329638 1.263258 2.879062 1.407246 5.218027   
## BsmtFinSF2 BsmtUnfSF X1stFlrSF X2ndFlrSF BsmtFullBath FullBath   
## 1.463436 4.022976 4.356945 4.453603 1.972001 2.854257   
## HalfBath BedroomAbvGr KitchenAbvGr Fireplaces GarageCars OpenPorchSF   
## 2.238842 1.687006 1.279141 1.509304 1.970249 1.181209   
## X3SsnPorch ScreenPorch PoolArea   
## 1.017618 1.078371 1.031503

which(as.vector(vif(n00))>treshold\_vif(summary(n00)$r.squared))# there are not highly correlated variables in the model

## integer(0)

# delete with a backwards stepwise : MasVnrArea, FullBath, HalfBath, BedroomAbvGr,OpenPorchSF, X3SsnPorch  
n0<- lm(log(SalePrice) ~ LotArea + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + BsmtFinSF2+ X1stFlrSF+ X2ndFlrSF +  
 BsmtFullBath + KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea, data = train\_new)

# Check BoxTidwell tests:   
# We guess that the variables with a higher range will have a logaritmic transformation (the area variables)  
  
n0<- lm(log(SalePrice) ~ LotArea + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + BsmtFinSF2+ X1stFlrSF+ X2ndFlrSF +  
 BsmtFullBath + KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea, data = train\_new)  
  
# we do BoxTidwall test to see which transformations we shall apply in the predictors to make a better model.   
# LotArea  
boxTidwell(log(SalePrice) ~ LotArea, ~OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + BsmtFinSF2+ X1stFlrSF+ X2ndFlrSF +  
 BsmtFullBath + KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea,data=train\_new, max.iter = 70 )

## MLE of lambda Score Statistic (z) Pr(>|z|)   
## -0.56093 -5.6351 1.749e-08 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## iterations = 9

v0<- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + BsmtFinSF2+ X1stFlrSF+ X2ndFlrSF +  
 BsmtFullBath + KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea, data = train\_new)  
# we shall convert the LotArea variable into logaritmic  
  
# X1stFlrSF  
boxTidwell(log(SalePrice) ~ X1stFlrSF,~ log(LotArea)+ OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + BsmtFinSF2+ X2ndFlrSF +  
 BsmtFullBath + KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea,data=train\_new, max.iter = 70 )

## MLE of lambda Score Statistic (z) Pr(>|z|)   
## 0.59555 -2.6347 0.008421 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## iterations = 2

v1<- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + BsmtFinSF2+ log(X1stFlrSF)+ X2ndFlrSF +  
 BsmtFullBath + KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea, data = train\_new)  
  
  
# X2stFlrSF  
  
boxTidwell(log(SalePrice) ~ I(X2ndFlrSF+0.01),~ log(LotArea)+ OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + BsmtFinSF2+X1stFlrSF +  
 BsmtFullBath + KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea,data=train\_new, max.iter = 70 ) # we dont need

## MLE of lambda Score Statistic (z) Pr(>|z|)  
## 0.98748 -0.1429 0.8864  
##   
## iterations = 2

boxTidwell(log(SalePrice) ~ I(PoolArea+0.01),~ log(LotArea)+ OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + BsmtFinSF2+X1stFlrSF + X2ndFlrSF+  
 BsmtFullBath + KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch,data=train\_new, max.iter = 70 ) # no either

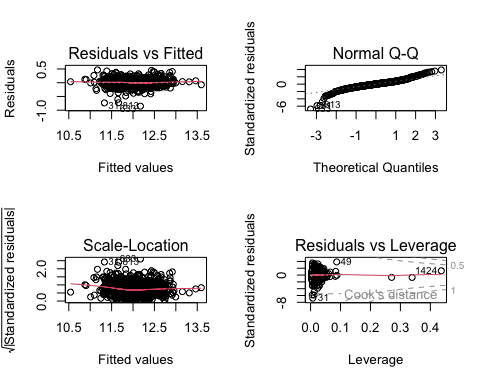
## Warning in boxTidwell.default(y, X1, X2, max.iter = max.iter, tol = tol, :  
## maximum iterations exceeded

## MLE of lambda Score Statistic (z) Pr(>|z|)   
## 20.465 1.711 0.08708 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## iterations = 71

#summary(v1)  
summary(v1)$r.squared

## [1] 0.9028157

par(mfrow=c(2,2))  
plot(v1)



par(mfrow=c(1,1))  
#influencePlot(v1)  
which(as.vector(vif(v1))>treshold\_vif(summary(v1)$r.squared))# there are not highly correlated variables in the model

## integer(0)

Therefore we have 2 potential models. We have been looking on the accuracy of the model, currently on 90% (will deppend on the seed), but we need to check if there is overfitting using the test sample.

We shall check

rmse <- function(fitted, actual){  
 sqrt(mean((fitted - actual)^2))  
}  
  
RSQUARE <- function(predictions, actual\_values) {  
 ss\_residual <- sum((actual\_values - predictions)^2)  
 ss\_total <- sum((actual\_values - mean(actual\_values))^2)  
 rsquare <- 1 - (ss\_residual / ss\_total)  
 return(rsquare)  
}  
  
  
results <- data.frame(Model = c("Model with train\_new sample",  
 "Model with test\_new sample"),  
 RSQRT = round(c(RSQUARE(exp(fitted(v1)), train\_new$SalePrice),  
 RSQUARE(exp(predict(v1, newdata = test\_new)), test\_new$SalePrice)),2),  
 RMSE = round(c(rmse(exp(fitted(v1)), train\_new$SalePrice),  
 rmse(exp(predict(v1, newdata = test\_new)), test\_new$SalePrice)),2))  
results

## Model RSQRT RMSE  
## 1 Model with train\_new sample 0.91 24295.81  
## 2 Model with test\_new sample 0.89 24796.11

Seems that the model has no overfitting. No significant difference on the measures of the model between the train sample and the test.

Now we shall proceed to adding the categorical variables for our model creation. We now add to the model

#names(train\_new)  
b0<- lm(log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + log(X1stFlrSF)+ X2ndFlrSF +  
 BsmtFullBath + FullBath + TotRmsAbvGrd+  
 KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea + Neighborhood  
 , data = train\_new)  
#summary(b0)  
summary(b0)$r.squared

## [1] 0.921046

anova(v1,b0)

## Analysis of Variance Table  
##   
## Model 1: log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond + YearBuilt +   
## BsmtFinSF1 + BsmtFinSF2 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## KitchenAbvGr + Fireplaces + GarageCars + ScreenPorch + PoolArea  
## Model 2: log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + Neighborhood  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 1002 15.973   
## 2 976 12.976 26 2.9962 8.6675 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(v1,b0)

## df AIC  
## v1 16 -1306.228  
## b0 42 -1465.503

treshold\_vif(summary(b0)$r.squared)

## [1] 12.6656

vif(b0) # High vif-> Highly correlated predictor -> wont add to the model

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.845403 1 1.358456  
## log(LotArea) 2.948867 1 1.717227  
## OverallQual 3.578430 1 1.891674  
## OverallCond 1.374806 1 1.172521  
## YearBuilt 6.309363 1 2.511845  
## BsmtFinSF1 2.234867 1 1.494947  
## log(X1stFlrSF) 4.450493 1 2.109619  
## X2ndFlrSF 4.659441 1 2.158574  
## BsmtFullBath 1.899193 1 1.378112  
## FullBath 2.539264 1 1.593507  
## TotRmsAbvGrd 4.118965 1 2.029523  
## KitchenAbvGr 1.453716 1 1.205702  
## Fireplaces 1.656721 1 1.287137  
## GarageCars 2.156018 1 1.468338  
## ScreenPorch 1.100605 1 1.049097  
## PoolArea 1.046065 1 1.022773  
## Neighborhood 47.377344 24 1.083697

b1<- lm(log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + log(X1stFlrSF)+ X2ndFlrSF +  
 BsmtFullBath + FullBath + TotRmsAbvGrd+  
 KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea +ExterQual  
 , data = train\_new)  
  
#summary(b1)  
summary(b1)$r.squared

## [1] 0.9081603

anova(n0,b1)

## Analysis of Variance Table  
##   
## Model 1: log(SalePrice) ~ LotArea + OverallQual + OverallCond + YearBuilt +   
## BsmtFinSF1 + BsmtFinSF2 + X1stFlrSF + X2ndFlrSF + BsmtFullBath +   
## KitchenAbvGr + Fireplaces + GarageCars + ScreenPorch + PoolArea  
## Model 2: log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + ExterQual  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 1002 16.557   
## 2 997 15.094 5 1.4631 19.328 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(n0,b1)

## df AIC  
## n0 16 -1269.667  
## b1 21 -1353.754

treshold\_vif(summary(b1)$r.squared)

## [1] 10.88853

vif(b1)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.740964 1 1.319456  
## log(LotArea) 1.935320 1 1.391158  
## OverallQual 3.554715 1 1.885395  
## OverallCond 1.289128 1 1.135398  
## YearBuilt 2.594381 1 1.610708  
## BsmtFinSF1 2.134056 1 1.460841  
## log(X1stFlrSF) 4.028248 1 2.007050  
## X2ndFlrSF 4.072475 1 2.018037  
## BsmtFullBath 1.835826 1 1.354927  
## FullBath 2.301964 1 1.517223  
## TotRmsAbvGrd 3.890964 1 1.972553  
## KitchenAbvGr 1.359861 1 1.166131  
## Fireplaces 1.485164 1 1.218673  
## GarageCars 1.999835 1 1.414155  
## ScreenPorch 1.077423 1 1.037990  
## PoolArea 1.029241 1 1.014515  
## ExterQual 2.921819 3 1.195663

# check interactions  
b11<- lm(log(SalePrice) ~ (LotFrontage + log(LotArea) + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + log(X1stFlrSF)+ X2ndFlrSF +  
 BsmtFullBath + FullBath + TotRmsAbvGrd+  
 KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea)\*ExterQual  
 , data = train\_new)  
#summary(b11)  
summary(b11)$r.squared

## [1] 0.9184602

anova(b1,b11) # better with interactions

## Analysis of Variance Table  
##   
## Model 1: log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + ExterQual  
## Model 2: log(SalePrice) ~ (LotFrontage + log(LotArea) + OverallQual +   
## OverallCond + YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF +   
## BsmtFullBath + FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces +   
## GarageCars + ScreenPorch + PoolArea) \* ExterQual  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 997 15.094   
## 2 958 13.401 39 1.6928 3.1029 1.362e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b1,b11)

## df AIC  
## b1 21 -1353.754  
## b11 60 -1396.730

b2<- lm(log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + log(X1stFlrSF)+ X2ndFlrSF +  
 BsmtFullBath + FullBath + TotRmsAbvGrd+  
 KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea +ExterQual + BsmtQual  
 , data = train\_new)  
  
#summary(b2)  
summary(b2)$r.squared

## [1] 0.9131766

anova(b1,b2)

## Analysis of Variance Table  
##   
## Model 1: log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + ExterQual  
## Model 2: log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + ExterQual + BsmtQual  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 997 15.094   
## 2 993 14.270 4 0.82446 14.343 2.175e-11 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b1,b2)

## df AIC  
## b1 21 -1353.754  
## b2 25 -1402.878

treshold\_vif(summary(b2)$r.squared)

## [1] 11.51764

vif(b2)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.750297 1 1.322988  
## log(LotArea) 1.948027 1 1.395717  
## OverallQual 3.810265 1 1.951990  
## OverallCond 1.314856 1 1.146672  
## YearBuilt 3.283505 1 1.812044  
## BsmtFinSF1 2.185455 1 1.478329  
## log(X1stFlrSF) 4.039657 1 2.009890  
## X2ndFlrSF 4.124406 1 2.030863  
## BsmtFullBath 1.849662 1 1.360023  
## FullBath 2.400428 1 1.549332  
## TotRmsAbvGrd 3.974461 1 1.993605  
## KitchenAbvGr 1.416237 1 1.190058  
## Fireplaces 1.490201 1 1.220738  
## GarageCars 2.021997 1 1.421969  
## ScreenPorch 1.081930 1 1.040159  
## PoolArea 1.030948 1 1.015356  
## ExterQual 3.702850 3 1.243816  
## BsmtQual 5.045907 4 1.224242

# interactions  
b21<- lm(log(SalePrice) ~ (LotFrontage + log(LotArea) + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + log(X1stFlrSF)+ X2ndFlrSF +  
 BsmtFullBath + FullBath + TotRmsAbvGrd+  
 KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea +ExterQual) \*BsmtQual  
 , data = train\_new)  
#summary(b21)  
summary(b21)$r.squared

## [1] 0.9223763

anova(b2,b21) # better with interaction

## Analysis of Variance Table  
##   
## Model 1: log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + ExterQual + BsmtQual  
## Model 2: log(SalePrice) ~ (LotFrontage + log(LotArea) + OverallQual +   
## OverallCond + YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF +   
## BsmtFullBath + FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces +   
## GarageCars + ScreenPorch + PoolArea + ExterQual) \* BsmtQual  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 993 14.270   
## 2 929 12.758 64 1.512 1.7203 0.0005471 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b2,b21)

## df AIC  
## b2 25 -1402.878  
## b21 89 -1388.785

anova(b11,b21) # but worse than b11

## Analysis of Variance Table  
##   
## Model 1: log(SalePrice) ~ (LotFrontage + log(LotArea) + OverallQual +   
## OverallCond + YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF +   
## BsmtFullBath + FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces +   
## GarageCars + ScreenPorch + PoolArea) \* ExterQual  
## Model 2: log(SalePrice) ~ (LotFrontage + log(LotArea) + OverallQual +   
## OverallCond + YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF +   
## BsmtFullBath + FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces +   
## GarageCars + ScreenPorch + PoolArea + ExterQual) \* BsmtQual  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 958 13.401   
## 2 929 12.758 29 0.64362 1.6161 0.02148 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

b3<- lm(log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + log(X1stFlrSF)+ X2ndFlrSF +  
 BsmtFullBath + FullBath + TotRmsAbvGrd+  
 KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea +ExterQual + BsmtQual + KitchenQual  
 , data = train\_new)  
  
#summary(b3)  
summary(b3)$r.squared

## [1] 0.9144211

anova(b2,b3)

## Analysis of Variance Table  
##   
## Model 1: log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + ExterQual + BsmtQual  
## Model 2: log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + ExterQual + BsmtQual + KitchenQual  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 993 14.270   
## 2 990 14.065 3 0.20454 4.7989 0.002524 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b2,b3)

## df AIC  
## b2 25 -1402.878  
## b3 28 -1411.561

treshold\_vif(summary(b3)$r.squared)

## [1] 11.68513

vif(b3)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.763950 1 1.328138  
## log(LotArea) 1.948867 1 1.396018  
## OverallQual 3.914671 1 1.978553  
## OverallCond 1.391745 1 1.179722  
## YearBuilt 3.382198 1 1.839075  
## BsmtFinSF1 2.206807 1 1.485533  
## log(X1stFlrSF) 4.063506 1 2.015814  
## X2ndFlrSF 4.141473 1 2.035061  
## BsmtFullBath 1.860763 1 1.364098  
## FullBath 2.431320 1 1.559269  
## TotRmsAbvGrd 4.000383 1 2.000096  
## KitchenAbvGr 1.427794 1 1.194904  
## Fireplaces 1.491987 1 1.221469  
## GarageCars 2.046528 1 1.430569  
## ScreenPorch 1.088365 1 1.043247  
## PoolArea 1.033201 1 1.016465  
## ExterQual 5.755142 3 1.338678  
## BsmtQual 5.530908 4 1.238368  
## KitchenQual 4.824372 3 1.299891

#with interactions  
b31<- lm(log(SalePrice) ~ (LotFrontage + log(LotArea) + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + log(X1stFlrSF)+ X2ndFlrSF +  
 BsmtFullBath + FullBath + TotRmsAbvGrd+  
 KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea +ExterQual + BsmtQual)\* KitchenQual  
 , data = train\_new)  
  
#summary(b31)  
summary(b31)$r.squared

## [1] 0.9266179

anova(b3,b31) # better model with interactions

## Analysis of Variance Table  
##   
## Model 1: log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + ExterQual + BsmtQual + KitchenQual  
## Model 2: log(SalePrice) ~ (LotFrontage + log(LotArea) + OverallQual +   
## OverallCond + YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF +   
## BsmtFullBath + FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces +   
## GarageCars + ScreenPorch + PoolArea + ExterQual + BsmtQual) \*   
## KitchenQual  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 990 14.065   
## 2 931 12.061 59 2.0046 2.6227 1.591e-09 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b3,b31)

## df AIC  
## b3 28 -1411.561  
## b31 87 -1449.932

b4<- lm(log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + log(X1stFlrSF)+ X2ndFlrSF +  
 BsmtFullBath + FullBath + TotRmsAbvGrd+  
 KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea +ExterQual + BsmtQual + KitchenQual+ GarageFinish  
 , data = train\_new)  
  
#summary(b4)  
summary(b4)$r.squared

## [1] 0.9146056

anova(b3,b4) # not better model -> we dont use this variable

## Analysis of Variance Table  
##   
## Model 1: log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + ExterQual + BsmtQual + KitchenQual  
## Model 2: log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + ExterQual + BsmtQual + KitchenQual +   
## GarageFinish  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 990 14.065   
## 2 987 14.035 3 0.030316 0.7107 0.5457

AIC(b3,b4)

## df AIC  
## b3 28 -1411.561  
## b4 31 -1407.755

treshold\_vif(summary(b4)$r.squared)

## [1] 11.71037

vif(b4)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.782719 1 1.335185  
## log(LotArea) 1.951510 1 1.396964  
## OverallQual 3.921986 1 1.980400  
## OverallCond 1.393772 1 1.180581  
## YearBuilt 3.633225 1 1.906102  
## BsmtFinSF1 2.215722 1 1.488530  
## log(X1stFlrSF) 4.090888 1 2.022594  
## X2ndFlrSF 4.156659 1 2.038789  
## BsmtFullBath 1.864202 1 1.365358  
## FullBath 2.485931 1 1.576684  
## TotRmsAbvGrd 4.055086 1 2.013724  
## KitchenAbvGr 1.458613 1 1.207731  
## Fireplaces 1.522185 1 1.233769  
## GarageCars 3.053902 1 1.747542  
## ScreenPorch 1.090138 1 1.044097  
## PoolArea 1.034588 1 1.017147  
## ExterQual 5.932326 3 1.345460  
## BsmtQual 5.804067 4 1.245852  
## KitchenQual 4.914286 3 1.303897  
## GarageFinish 3.942767 3 1.256898

b5<- lm(log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + log(X1stFlrSF)+ X2ndFlrSF +  
 BsmtFullBath + FullBath + TotRmsAbvGrd+  
 KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea +ExterQual + BsmtQual + KitchenQual + FireplaceQu  
 , data = train\_new)  
  
#summary(b5)  
summary(b5)$r.squared

## [1] 0.9151697

anova(b3,b5) # not better model

## Analysis of Variance Table  
##   
## Model 1: log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + ExterQual + BsmtQual + KitchenQual  
## Model 2: log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + ExterQual + BsmtQual + KitchenQual +   
## FireplaceQu  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 990 14.065   
## 2 985 13.942 5 0.12303 1.7384 0.1231

AIC(b3,b5)

## df AIC  
## b3 28 -1411.561  
## b5 33 -1410.496

treshold\_vif(summary(b5)$r.squared)

## [1] 11.78824

vif(b5)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.771465 1 1.330964  
## log(LotArea) 1.961075 1 1.400384  
## OverallQual 3.959324 1 1.989805  
## OverallCond 1.394117 1 1.180727  
## YearBuilt 3.450442 1 1.857537  
## BsmtFinSF1 2.241445 1 1.497145  
## log(X1stFlrSF) 4.141760 1 2.035131  
## X2ndFlrSF 4.262039 1 2.064471  
## BsmtFullBath 1.877039 1 1.370051  
## FullBath 2.451467 1 1.565716  
## TotRmsAbvGrd 4.072611 1 2.018071  
## KitchenAbvGr 1.443718 1 1.201548  
## Fireplaces 5.990908 1 2.447633  
## GarageCars 2.054623 1 1.433396  
## ScreenPorch 1.094968 1 1.046407  
## PoolArea 1.047875 1 1.023657  
## ExterQual 5.827002 3 1.341449  
## BsmtQual 5.949713 4 1.249718  
## KitchenQual 5.038032 3 1.309313  
## FireplaceQu 9.229865 5 1.248877

b6<- lm(log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + log(X1stFlrSF)+ X2ndFlrSF +  
 BsmtFullBath + FullBath + TotRmsAbvGrd+  
 KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea +ExterQual + BsmtQual + KitchenQual+Foundation  
 , data = train\_new)  
  
#summary(b6)  
summary(b6)$r.squared

## [1] 0.9166439

anova(b3,b6)

## Analysis of Variance Table  
##   
## Model 1: log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + ExterQual + BsmtQual + KitchenQual  
## Model 2: log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + ExterQual + BsmtQual + KitchenQual +   
## Foundation  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 990 14.065   
## 2 985 13.700 5 0.36533 5.2533 9.078e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b3,b6)

## df AIC  
## b3 28 -1411.561  
## b6 33 -1428.325

treshold\_vif(summary(b6)$r.squared)

## [1] 11.99673

vif(b6) # we dont add it because highly correlated

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.828715 1 1.352300  
## log(LotArea) 1.957237 1 1.399013  
## OverallQual 4.007396 1 2.001848  
## OverallCond 1.451373 1 1.204729  
## YearBuilt 5.034807 1 2.243838  
## BsmtFinSF1 2.247900 1 1.499300  
## log(X1stFlrSF) 4.100633 1 2.025002  
## X2ndFlrSF 4.188673 1 2.046625  
## BsmtFullBath 1.866171 1 1.366079  
## FullBath 2.465051 1 1.570048  
## TotRmsAbvGrd 4.017567 1 2.004387  
## KitchenAbvGr 1.435622 1 1.198175  
## Fireplaces 1.524888 1 1.234863  
## GarageCars 2.081484 1 1.442735  
## ScreenPorch 1.090213 1 1.044133  
## PoolArea 1.036614 1 1.018143  
## ExterQual 5.970564 3 1.346902  
## BsmtQual 20.910648 4 1.462332  
## KitchenQual 5.119344 3 1.312812  
## Foundation 17.831310 5 1.333885

b7<- lm(log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + log(X1stFlrSF)+ X2ndFlrSF +  
 BsmtFullBath + FullBath + TotRmsAbvGrd+  
 KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea +ExterQual + BsmtQual + KitchenQual+GarageType  
 , data = train\_new)  
  
#summary(b7)  
summary(b7)$r.squared

## [1] 0.9151455

anova(b3,b7)

## Analysis of Variance Table  
##   
## Model 1: log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + ExterQual + BsmtQual + KitchenQual  
## Model 2: log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + ExterQual + BsmtQual + KitchenQual +   
## GarageType  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 990 14.065   
## 2 984 13.946 6 0.11905 1.3999 0.2115

AIC(b3,b7)

## df AIC  
## b3 28 -1411.561  
## b7 34 -1408.205

treshold\_vif(summary(b7)$r.squared)

## [1] 11.78487

vif(b7)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.800942 1 1.341992  
## log(LotArea) 1.972709 1 1.404531  
## OverallQual 3.974310 1 1.993567  
## OverallCond 1.400483 1 1.183420  
## YearBuilt 4.011927 1 2.002979  
## BsmtFinSF1 2.221693 1 1.490534  
## log(X1stFlrSF) 4.146020 1 2.036178  
## X2ndFlrSF 4.341041 1 2.083517  
## BsmtFullBath 1.865310 1 1.365764  
## FullBath 2.466049 1 1.570366  
## TotRmsAbvGrd 4.052246 1 2.013019  
## KitchenAbvGr 1.534963 1 1.238936  
## Fireplaces 1.528121 1 1.236172  
## GarageCars 3.202586 1 1.789577  
## ScreenPorch 1.095980 1 1.046891  
## PoolArea 1.035043 1 1.017370  
## ExterQual 5.994957 3 1.347817  
## BsmtQual 5.858709 4 1.247312  
## KitchenQual 5.060148 3 1.310269  
## GarageType 5.044076 6 1.144367

# interactions   
b71<- lm(log(SalePrice) ~ (LotFrontage + log(LotArea) + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + log(X1stFlrSF)+ X2ndFlrSF +  
 BsmtFullBath + FullBath + TotRmsAbvGrd+  
 KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea)\*GarageType +ExterQual + BsmtQual + KitchenQual  
 , data = train\_new)  
  
#summary(b71)  
summary(b71)$r.squared

## [1] 0.9280067

anova(b7,b71) # better model

## Analysis of Variance Table  
##   
## Model 1: log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + ExterQual + BsmtQual + KitchenQual +   
## GarageType  
## Model 2: log(SalePrice) ~ (LotFrontage + log(LotArea) + OverallQual +   
## OverallCond + YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF +   
## BsmtFullBath + FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces +   
## GarageCars + ScreenPorch + PoolArea) \* GarageType + ExterQual +   
## BsmtQual + KitchenQual  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 984 13.946   
## 2 923 11.832 61 2.1138 2.7031 2.49e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b7,b71)

## df AIC  
## b7 34 -1408.205  
## b71 95 -1453.365

treshold\_vif(summary(b71)$r.squared)

## [1] 13.89018

b8<- lm(log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + log(X1stFlrSF)+ X2ndFlrSF +  
 BsmtFullBath + FullBath + TotRmsAbvGrd+  
 KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea +ExterQual + BsmtQual + KitchenQual+GarageType+MSSubClass  
 , data = train\_new)  
  
#summary(b8)  
summary(b8)$r.squared

## [1] 0.9171144

anova(b7,b8) # not better model

## Analysis of Variance Table  
##   
## Model 1: log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + ExterQual + BsmtQual + KitchenQual +   
## GarageType  
## Model 2: log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + ExterQual + BsmtQual + KitchenQual +   
## GarageType + MSSubClass  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 984 13.946   
## 2 970 13.623 14 0.32361 1.6459 0.06169 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b7,b8)

## df AIC  
## b7 34 -1408.205  
## b8 48 -1404.082

treshold\_vif(summary(b8)$r.squared)

## [1] 12.06483

vif(b8) # highly correlated

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.937765 1 1.392036  
## log(LotArea) 2.864402 1 1.692455  
## OverallQual 4.265604 1 2.065334  
## OverallCond 1.479248 1 1.216243  
## YearBuilt 8.922006 1 2.986973  
## BsmtFinSF1 2.258019 1 1.502670  
## log(X1stFlrSF) 4.807780 1 2.192665  
## X2ndFlrSF 8.886468 1 2.981018  
## BsmtFullBath 1.928847 1 1.388829  
## FullBath 2.547005 1 1.595934  
## TotRmsAbvGrd 4.304529 1 2.074736  
## KitchenAbvGr 3.518681 1 1.875815  
## Fireplaces 1.608958 1 1.268447  
## GarageCars 3.264196 1 1.806709  
## ScreenPorch 1.111963 1 1.054497  
## PoolArea 1.059153 1 1.029151  
## ExterQual 6.345423 3 1.360641  
## BsmtQual 7.535867 4 1.287187  
## KitchenQual 5.500766 3 1.328629  
## GarageType 6.638012 6 1.170855  
## MSSubClass 483.369912 14 1.247002

b9<- lm(log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + log(X1stFlrSF)+ X2ndFlrSF +  
 BsmtFullBath + FullBath + TotRmsAbvGrd+  
 KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea +ExterQual + BsmtQual + KitchenQual+GarageType+BsmtFinType1  
 , data = train\_new)  
  
#summary(b9)  
summary(b9)$r.squared

## [1] 0.9153663

anova(b7,b9) # not better model

## Analysis of Variance Table  
##   
## Model 1: log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + ExterQual + BsmtQual + KitchenQual +   
## GarageType  
## Model 2: log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + ExterQual + BsmtQual + KitchenQual +   
## GarageType + BsmtFinType1  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 984 13.946   
## 2 979 13.910 5 0.036298 0.5109 0.7681

AIC(b7,b9)

## df AIC  
## b7 34 -1408.205  
## b9 39 -1400.855

treshold\_vif(summary(b9)$r.squared)

## [1] 11.81563

AIC(b7,b71,b3,b31,b2,b21,b1,b11) # b31 better but too many df, -> we keep b3

## df AIC  
## b7 34 -1408.205  
## b71 95 -1453.365  
## b3 28 -1411.561  
## b31 87 -1449.932  
## b2 25 -1402.878  
## b21 89 -1388.785  
## b1 21 -1353.754  
## b11 60 -1396.730

# now we have done a forward stepwise, we can see if we need all the variables doing a backwards stepwise, we can see that in the model: LotFrontage PoolArea and GarageType are less influental, for better explainability we can not consider it in our model.  
final\_model<-b3

results <- data.frame(Model = c("Model with train\_new sample",  
 "Model with test\_new sample"),  
 RSQRT = round(c(RSQUARE(exp(fitted(final\_model)), train\_new$SalePrice),  
 RSQUARE(exp(predict(final\_model, newdata = test\_new)), test\_new$SalePrice)),2),  
 RMSE = round(c(rmse(exp(fitted(final\_model)), train\_new$SalePrice),  
 rmse(exp(predict(final\_model, newdata = test\_new)), test\_new$SalePrice)),2))  
results

## Model RSQRT RMSE  
## 1 Model with train\_new sample 0.93 22146.29  
## 2 Model with test\_new sample 0.91 22901.87

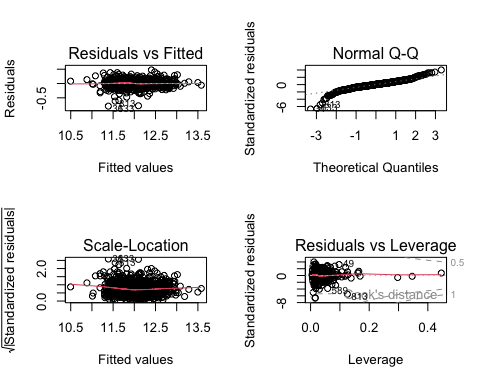
## interpret the model

The Residuals vs Fitted plot shows that the residuals follow a good linear pattern, which meets the regression assumptions very well. The Normal Q-Q plot shows that the standard errors are mostly normally distributed with a small amount of deviating prediction at the upper end of the tail and the beginning of the tail. The scale-location plot shows that homoscedasticity is satisfied as a nearly straight line is obtained. The residuals vs leverage plot shows that our model includes some high-leverage (so highly influential in our model) points that deviate significantly (more than 4 standardized residuals away) and asymmetrically from the prediction. An influence plot further confirms this believe

summary(final\_model)

##   
## Call:  
## lm(formula = log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual +   
## OverallCond + YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF +   
## BsmtFullBath + FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces +   
## GarageCars + ScreenPorch + PoolArea + ExterQual + BsmtQual +   
## KitchenQual, data = train\_new)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.79113 -0.05879 0.00553 0.06448 0.47527   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.588e+00 5.101e-01 5.073 4.68e-07 \*\*\*  
## LotFrontage -4.368e-05 2.161e-04 -0.202 0.839866   
## log(LotArea) 7.703e-02 1.049e-02 7.343 4.37e-13 \*\*\*  
## OverallQual 5.984e-02 5.268e-03 11.359 < 2e-16 \*\*\*  
## OverallCond 4.858e-02 3.951e-03 12.298 < 2e-16 \*\*\*  
## YearBuilt 2.741e-03 2.267e-04 12.092 < 2e-16 \*\*\*  
## BsmtFinSF1 7.827e-05 1.270e-05 6.165 1.03e-09 \*\*\*  
## log(X1stFlrSF) 3.858e-01 2.412e-02 15.997 < 2e-16 \*\*\*  
## X2ndFlrSF 2.325e-04 1.744e-05 13.328 < 2e-16 \*\*\*  
## BsmtFullBath 2.622e-02 9.926e-03 2.642 0.008377 \*\*   
## FullBath 1.786e-02 1.075e-02 1.661 0.096951 .   
## TotRmsAbvGrd 6.259e-03 4.618e-03 1.355 0.175599   
## KitchenAbvGr -8.064e-02 2.001e-02 -4.029 6.02e-05 \*\*\*  
## Fireplaces 2.670e-02 7.054e-03 3.785 0.000163 \*\*\*  
## GarageCars 4.809e-02 6.956e-03 6.914 8.46e-12 \*\*\*  
## ScreenPorch 2.577e-04 7.373e-05 3.495 0.000494 \*\*\*  
## PoolArea 1.095e-04 1.076e-04 1.018 0.308973   
## ExterQualFa -1.733e-01 5.132e-02 -3.377 0.000762 \*\*\*  
## ExterQualGd -2.533e-02 2.535e-02 -1.000 0.317782   
## ExterQualTA -7.250e-02 2.811e-02 -2.579 0.010055 \*   
## BsmtQualFa -1.635e-01 3.369e-02 -4.854 1.41e-06 \*\*\*  
## BsmtQualGd -9.679e-02 1.875e-02 -5.161 2.96e-07 \*\*\*  
## BsmtQualNo Basement -2.272e-01 3.339e-02 -6.806 1.74e-11 \*\*\*  
## BsmtQualTA -1.186e-01 2.181e-02 -5.437 6.82e-08 \*\*\*  
## KitchenQualFa -7.639e-02 3.370e-02 -2.267 0.023624 \*   
## KitchenQualGd -4.612e-02 1.922e-02 -2.400 0.016591 \*   
## KitchenQualTA -7.675e-02 2.123e-02 -3.614 0.000316 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1192 on 990 degrees of freedom  
## Multiple R-squared: 0.9144, Adjusted R-squared: 0.9122   
## F-statistic: 406.9 on 26 and 990 DF, p-value: < 2.2e-16

par(mfrow=c(2,2))  
plot(final\_model)



coefficients <- coef(final\_model)  
coefficients

## (Intercept) LotFrontage log(LotArea) OverallQual   
## 2.587505e+00 -4.368028e-05 7.702995e-02 5.983627e-02   
## OverallCond YearBuilt BsmtFinSF1 log(X1stFlrSF)   
## 4.858458e-02 2.741224e-03 7.827331e-05 3.858215e-01   
## X2ndFlrSF BsmtFullBath FullBath TotRmsAbvGrd   
## 2.324556e-04 2.622143e-02 1.785724e-02 6.258839e-03   
## KitchenAbvGr Fireplaces GarageCars ScreenPorch   
## -8.064418e-02 2.669999e-02 4.809421e-02 2.577251e-04   
## PoolArea ExterQualFa ExterQualGd ExterQualTA   
## 1.094828e-04 -1.732981e-01 -2.533387e-02 -7.250133e-02   
## BsmtQualFa BsmtQualGd BsmtQualNo Basement BsmtQualTA   
## -1.635415e-01 -9.679155e-02 -2.272070e-01 -1.185926e-01   
## KitchenQualFa KitchenQualGd KitchenQualTA   
## -7.638658e-02 -4.612331e-02 -7.674943e-02

We can see the coefficient with largest value (in absolute value) is log(X1stFlrSF) with positive impact, 0.385. Meaning that with all other things being equal(c.p) when the first floor square feet increases, the log-price also increases. Also, the level “BsmtQualNo Basement” has negative impact (-0.227). Meaning that the houses that have no basement will have a decrease on the sales price (c.p).