Assignment 1

Alicia Chimeno & Silvia Ferrer

2023-11-19

# 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 and Preproces the dataset

We start by looking at an overview of our 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)

## Preprocess and select variables

### 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), therefore 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

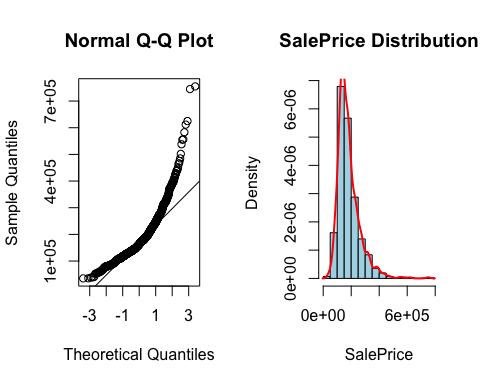
shapiro.test(train$SalePrice)

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

par(mfrow=c(1,2))  
shapiro.test(log(train$SalePrice)) # log-normality

##   
## Shapiro-Wilk normality test  
##   
## data: log(train$SalePrice)  
## W = 0.99121, p-value = 1.149e-07

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)



par(mfrow=c(1,1))

## Selection of 10 factor variables and numerical variables

We select the first 10 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 (EDA and univariate outliers)

## Analysis functions

To proceed with the predictor analysis, we do create the following functions for analyzing our numerical features:

## 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)  
}

We continue performing a descriptive analysis of each variable. Therefore, we are using the function update\_outliers\_count where we are updating a new variable named univ\_outl\_count, which is a count of the total univariate outliers an instance has, in order to track the data quality and associate having univariate outliers to some features with a correlation matrix. The order of this descriptive analysis follows the criteria of splitting into numerical and then categorical variables.

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)

## [1] "Id" "LotFrontage" "LotArea" "OverallQual"   
## [5] "OverallCond" "YearBuilt" "YearRemodAdd" "MasVnrArea"   
## [9] "BsmtFinSF1" "BsmtFinSF2" "BsmtUnfSF" "TotalBsmtSF"   
## [13] "X1stFlrSF" "X2ndFlrSF" "LowQualFinSF" "GrLivArea"   
## [17] "BsmtFullBath" "BsmtHalfBath" "FullBath" "HalfBath"   
## [21] "BedroomAbvGr" "KitchenAbvGr" "TotRmsAbvGrd" "Fireplaces"   
## [25] "GarageYrBlt" "GarageCars" "GarageArea" "WoodDeckSF"   
## [29] "OpenPorchSF" "EnclosedPorch" "X3SsnPorch" "ScreenPorch"   
## [33] "PoolArea" "MiscVal" "YrSold" "SalePrice"   
## [37] "Neighborhood" "ExterQual" "BsmtQual" "KitchenQual"   
## [41] "GarageFinish" "FireplaceQu" "Foundation" "GarageType"   
## [45] "MSSubClass" "BsmtFinType1" "univ\_outl\_count"

## Numerical variables

Firstly, we are analyzing the numerical variables, excluding the variable ‘Id’, which is not considered for the analysis. We distinguish between continuous variables (areas,) and discrete variables (nummber of garages, rooms…)

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

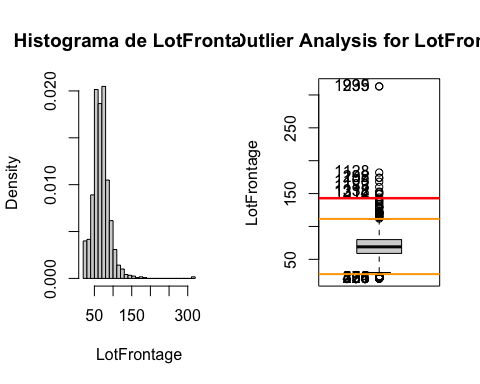
par(mfrow=c(1,2))  
numeric\_description(train2$LotFrontage, 30) # not normal distribution

## 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)

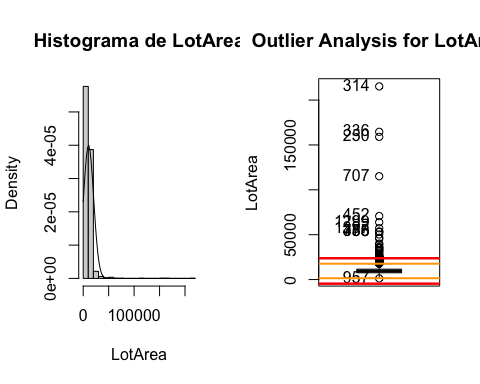
par(mfrow=c(1,2))  
numeric\_description(train2$LotArea, 20) # not normal distribution

## 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

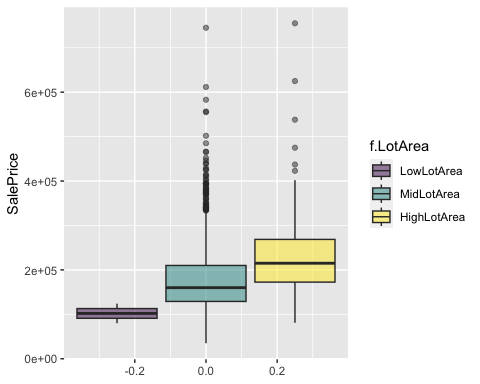


## Number of severe outliers: 34

train2 <- update\_outliers\_count(train2, "LotArea")  
  
# new variable:  
sm<-summary(train2$LotArea)  
plot(train2$LotArea,train2$SalePrice)  
  
iqr <- sm["3rd Qu."] - sm["1st Qu."]  
mild\_ub <- sm["3rd Qu."] + 1.5 \* iqr  
mild\_lb <- sm["1st Qu."] - 1.5 \* iqr  
  
train2$f.LotArea <- ifelse(  
 train2$LotArea <= mild\_lb,   
 1,   
 ifelse(  
 train2$LotArea > mild\_lb & train2$LotArea < mild\_ub,   
 2,  
 ifelse(  
 train2$LotArea >= mild\_ub,   
 3,  
   
 NA   
 )  
 )  
)  
  
train2$f.LotArea<- factor(train2$f.LotArea, labels=c("LowLotArea","MidLotArea","HighLotArea"), order = T, levels=c(1,2,3))  
table(train2$f.LotArea)

##   
## LowLotArea MidLotArea HighLotArea   
## 2 1391 67

ggplot(train2, aes(y=SalePrice, fill=f.LotArea)) + geom\_boxplot(alpha=0.5)



### “OverallQual” (cont)

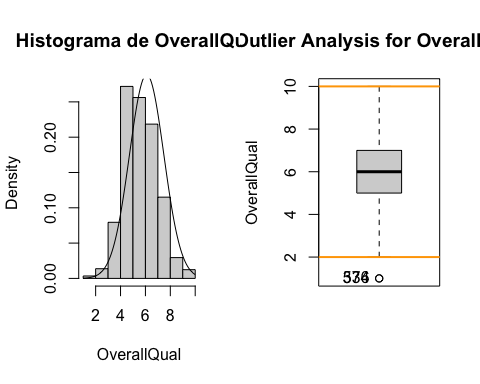
par(mfrow=c(1,2))  
numeric\_description(train2$OverallQual, 10) # not normal distribution

## 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)

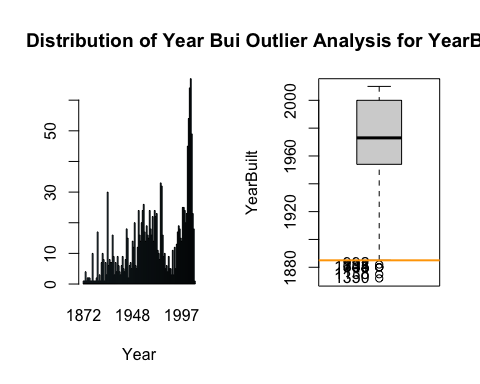
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1872 1954 1973 1971 2000 2010

par(mfrow=c(1,2))  
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

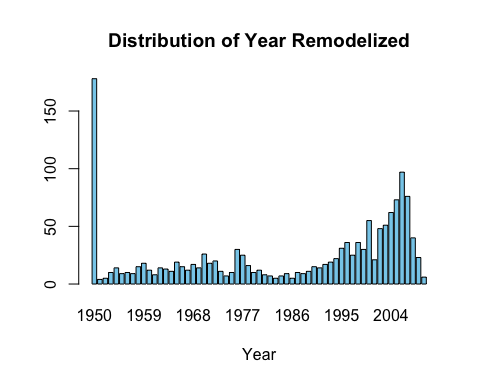
train2 <- update\_outliers\_count(train2, "YearBuilt")

### “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")

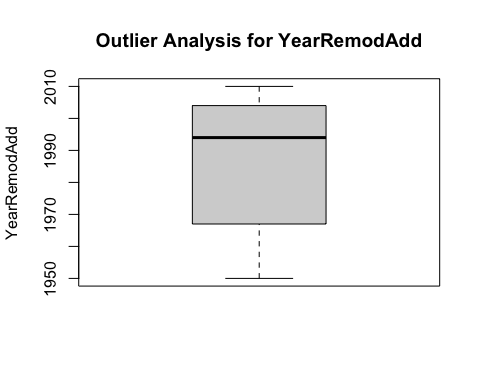


sum(is.na(train2$YearRemodAdd))

## [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)

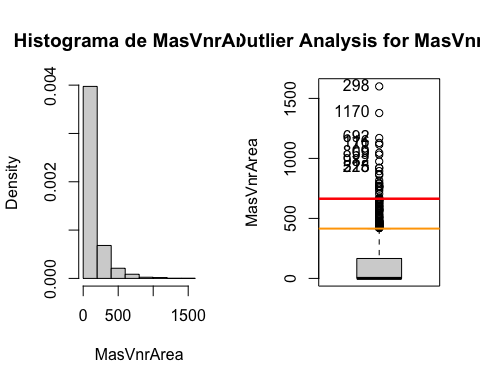
par(mfrow=c(1,2))  
numeric\_description(train2$MasVnrArea, 10)# not normal distribution

## 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")  
# find values that are NA instead of 0  
# if it has a material but a 0 in masvnrarea is a NA  
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)

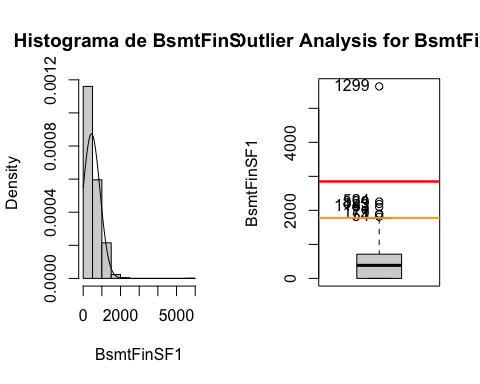
par(mfrow=c(1,2))  
numeric\_description(train2$BsmtFinSF1, 10)# not normal distribution

## 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)

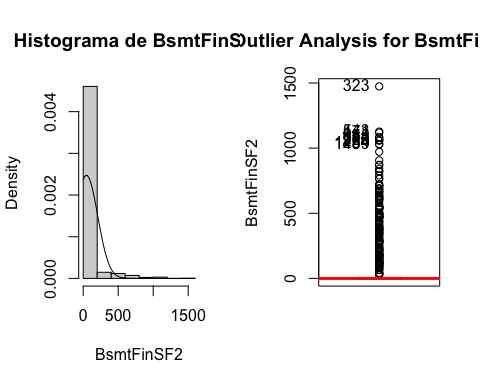
par(mfrow=c(1,2))  
numeric\_description(train2$BsmtFinSF2, 10) # not normal distribution

## 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)

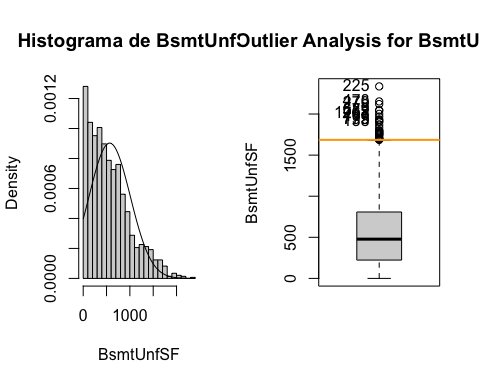
par(mfrow=c(1,2))  
numeric\_description(train2$BsmtUnfSF, 20) # not normal distribution

## 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)

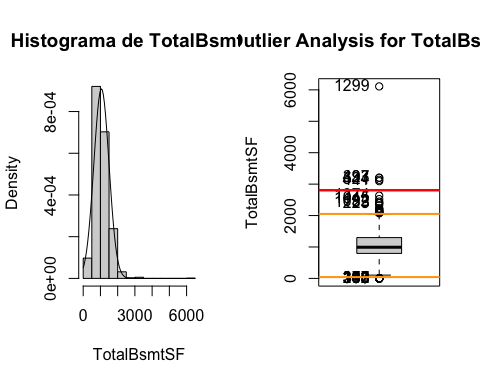
par(mfrow=c(1,2))  
numeric\_description(train2$TotalBsmtSF, 10)# not normal distribution

## 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)

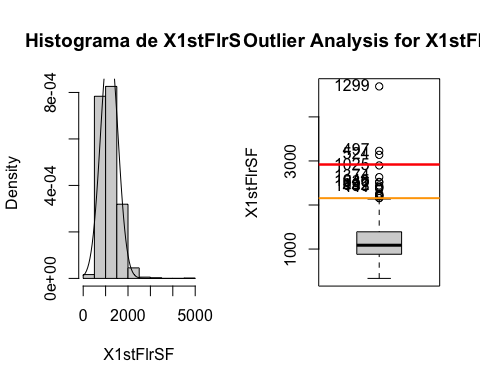
par(mfrow=c(1,2))  
numeric\_description(train2$X1stFlrSF, 10) # not normal distribution

## 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

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)

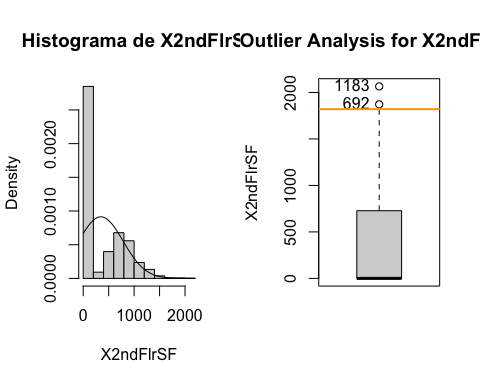
par(mfrow=c(1,2))  
numeric\_description(train2$X2ndFlrSF, 10)# not normal distribution

## 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)

par(mfrow=c(1,2))  
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

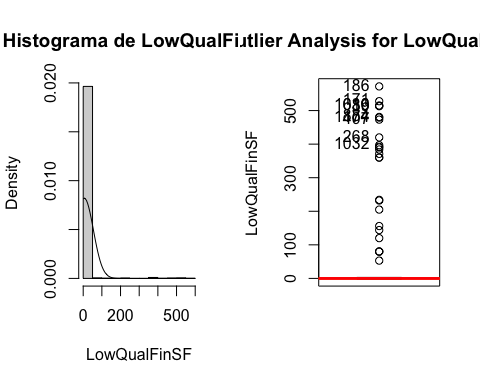
##   
## Shapiro-Wilk normality test  
##   
## data: variable  
## W = 0.09799, p-value < 2.2e-16

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

analyze\_outliers(train2,"LowQualFinSF")

## Number of mild outliers: 26



## Number of severe outliers: 26

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

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

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

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

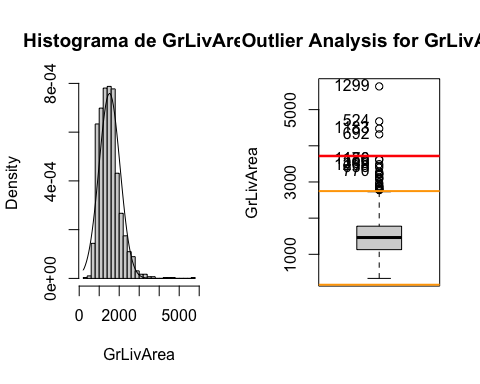
par(mfrow=c(1,2))  
numeric\_description(train2$GrLivArea, 20) # not normal distribution

## 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

par(mfrow=c(1,2))  
analyze\_outliers(train2,"BsmtFullBath")

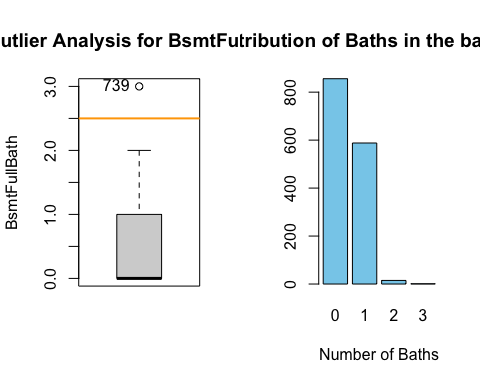
## Number of mild outliers: 1

## Number of severe outliers: 0

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

## [1] 0

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



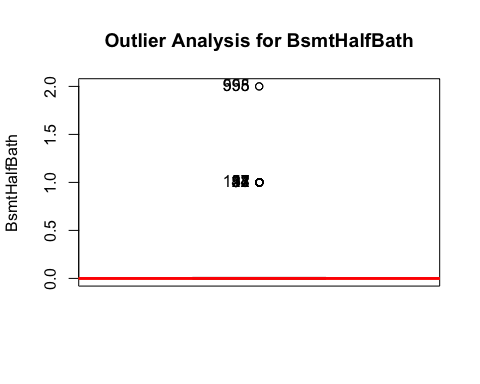
### BsmtHalfBath - # half baths in the basement (int)

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

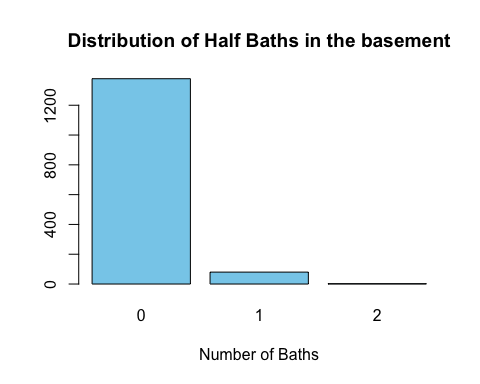
nearZeroVar(train2$BsmtHalfBath, saveMetrics = TRUE) # Variable not with near zero variance

## 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

par(mfrow=c(1,2))  
analyze\_outliers(train2,"FullBath")

## Number of mild outliers: 0

## Number of severe outliers: 0

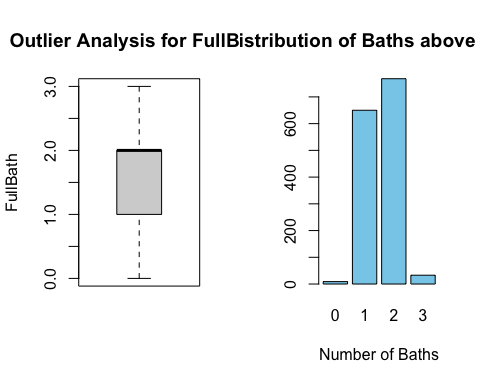
train2 <- update\_outliers\_count(train2, "FullBath")  
table(train2$FullBath)

##   
## 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

par(mfrow=c(1,2))  
analyze\_outliers(train2,"HalfBath")

## Number of mild outliers: 0

## Number of severe outliers: 0

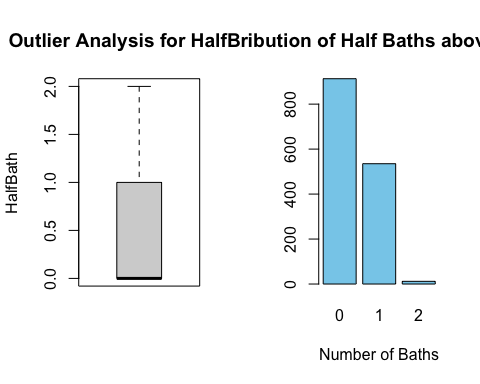
train2 <- update\_outliers\_count(train2, "HalfBath")  
table(train2$HalfBath)

##   
## 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

par(mfrow=c(1,2))  
analyze\_outliers(train2,"BedroomAbvGr")

## Number of mild outliers: 35

## Number of severe outliers: 1

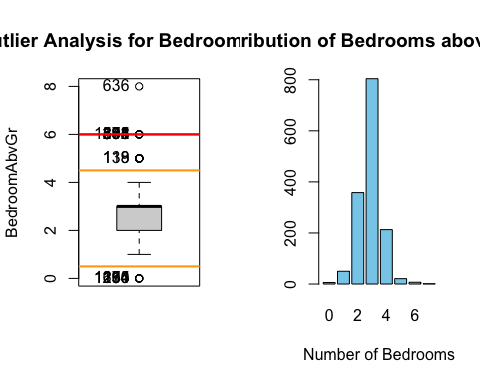
train2 <- update\_outliers\_count(train2, "BedroomAbvGr")  
table(train2$BedroomAbvGr)

##   
## 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

par(mfrow=c(1,2))  
analyze\_outliers(train2,"KitchenAbvGr")

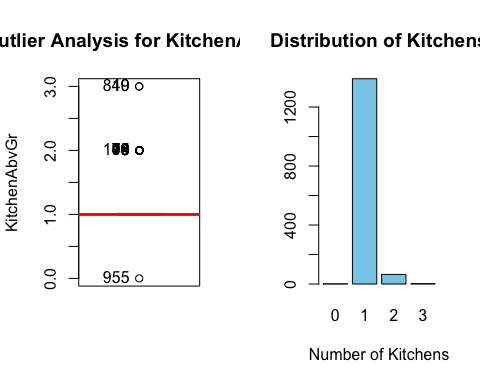
## Number of mild outliers: 68

## Number of severe outliers: 68

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

## [1] 0

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



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

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

par(mfrow=c(1,2))  
summary(train2$TotRmsAbvGrd)

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

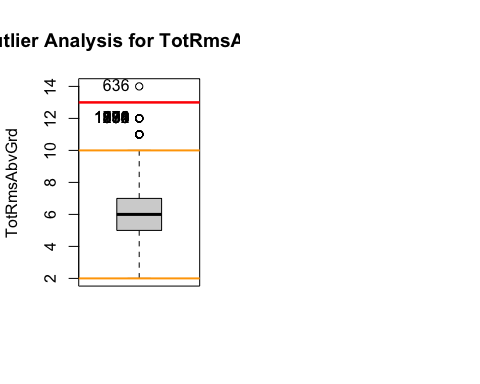
analyze\_outliers(train2,"TotRmsAbvGrd")

## 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

par(mfrow=c(1,2))  
#table(train2$Fireplaces)  
sum(is.na(train2$Fireplaces))

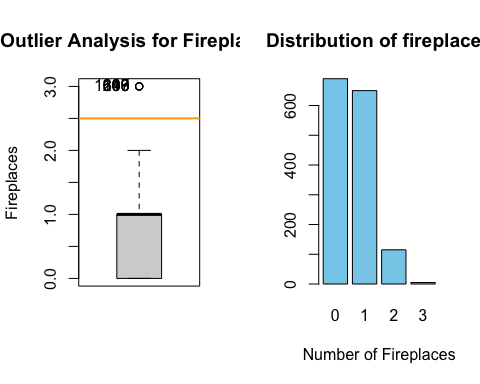
## [1] 0

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

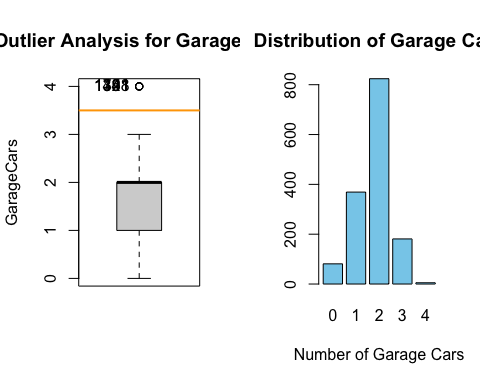
## [1] 0

par(mfrow=c(1,2))  
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)

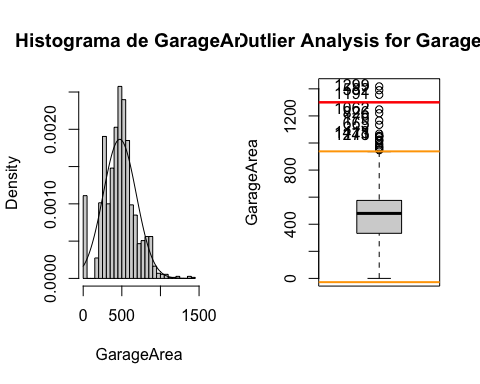
par(mfrow=c(1,2))  
numeric\_description(train2$GarageArea, 30) # not normal distribution

## 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

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

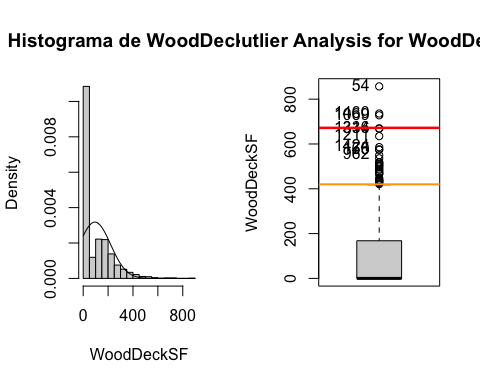
par(mfrow=c(1,2))  
numeric\_description(train2$WoodDeckSF, 30) # not normal distribution

## 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



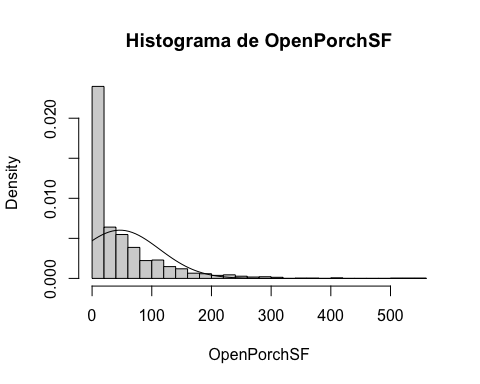
## Number of severe outliers: 3

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

### “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

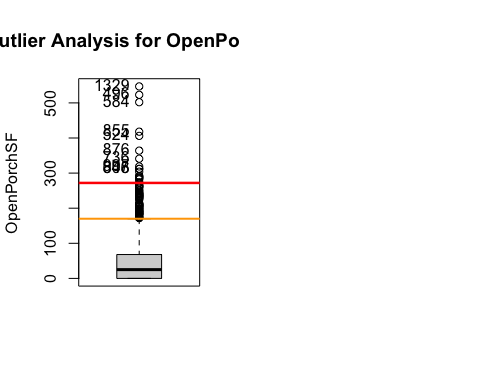
par(mfrow=c(1,2))  
analyze\_outliers(train2,"OpenPorchSF")

## Number of mild outliers: 77

## Number of severe outliers: 18

train2 <- update\_outliers\_count(train2, "OpenPorchSF")  
#table(train2$OpenPorchSF) # very centered data in 0  
train2$OpenPorch\_binary<-0  
train2$OpenPorch\_binary[which(train2$OpenPorchSF!=0)]<-1  
train2$OpenPorch\_binary<-as.factor(train2$OpenPorch\_binary)  
table(train2$OpenPorch\_binary)

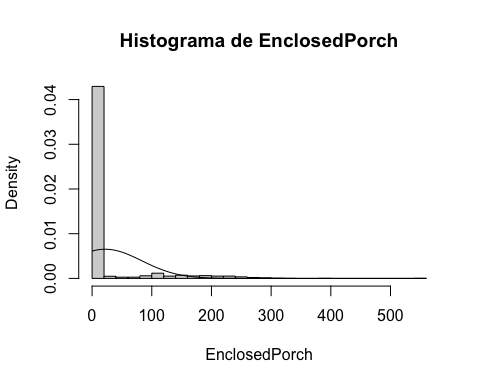
##   
## 0 1   
## 656 804



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

numeric\_description(train2$EnclosedPorch, 30)

## Summary:  
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.00 0.00 0.00 21.95 0.00 552.00   
##   
## Count of missing values: 0



##   
## Shapiro-Wilk normality test  
##   
## data: variable  
## W = 0.41444, p-value < 2.2e-16

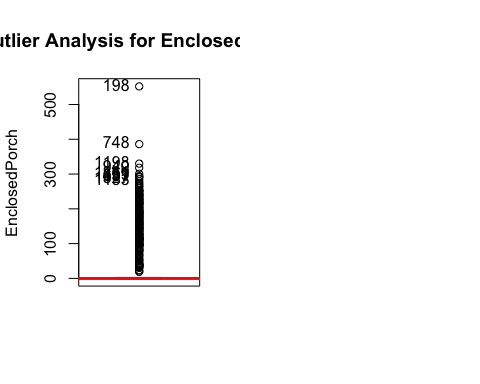
par(mfrow=c(1,2))  
analyze\_outliers(train2,"EnclosedPorch")

## Number of mild outliers: 208

## Number of severe outliers: 208

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

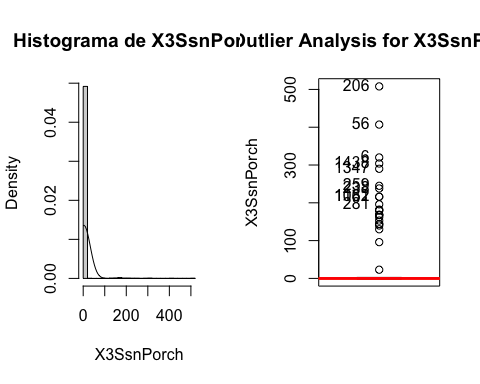
par(mfrow=c(1,2))  
numeric\_description(train2$X3SsnPorch, 30)

## 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)

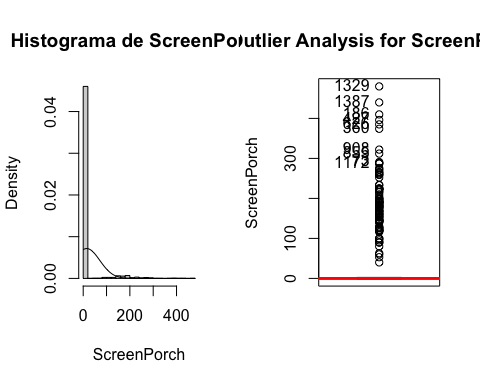
par(mfrow=c(1,2))  
numeric\_description(train2$ScreenPorch, 30) # not normally distributed

## 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)

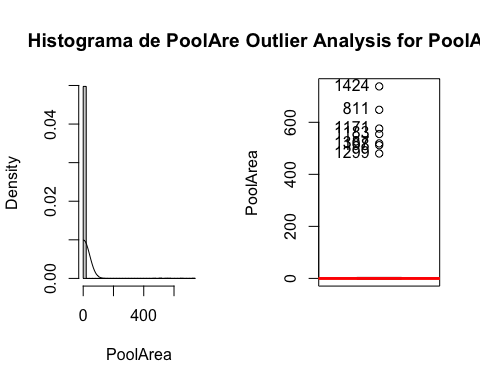
par(mfrow=c(1,2))  
numeric\_description(train2$PoolArea, 30)# not normally distributed

## 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

#table(train2$PoolArea) # very centered in 0 -> dont have pool  
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)

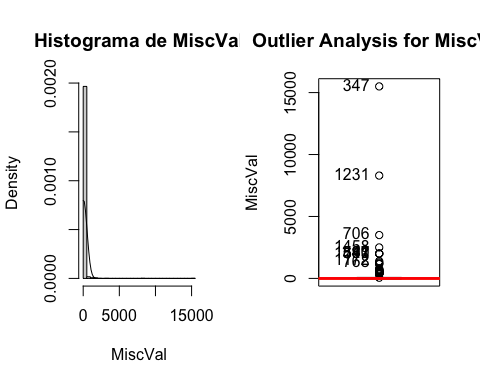
par(mfrow=c(1,2))  
numeric\_description(train2$MiscVal, 30) # not normally distributed

## 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

#table(train2$MiscVal) # very centered in 0   
analyze\_outliers(train2,"MiscVal")

## Number of mild outliers: 52



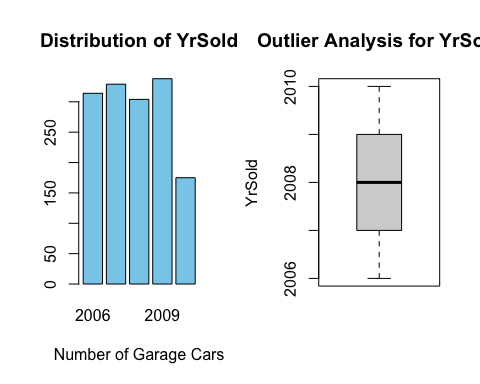
## Number of severe outliers: 52

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

### YrSold - (int)

par(mfrow=c(1,2))  
barplot(table(train2$YrSold), main = "Distribution of YrSold",xlab = "Number of Garage Cars",col = "skyblue")  
analyze\_outliers(train2,"YrSold")

## Number of mild outliers: 0



## Number of severe outliers: 0

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

## Categorical Variables

Categorical variables are analyzed.

### Neighborhood

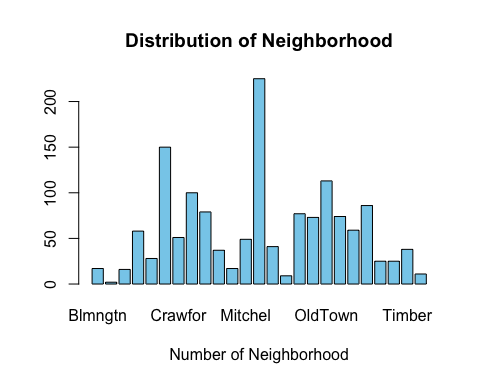
sum(is.na(train2$Neighborhood)) #0

## [1] 0

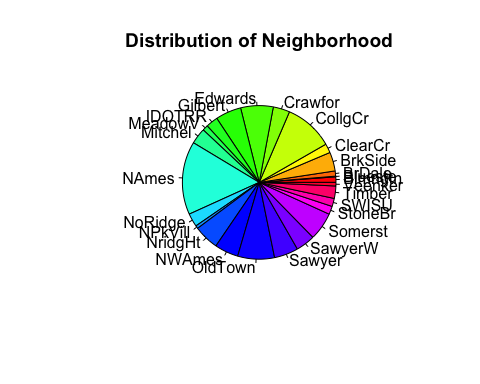
table(train2$Neighborhood)

##   
## 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

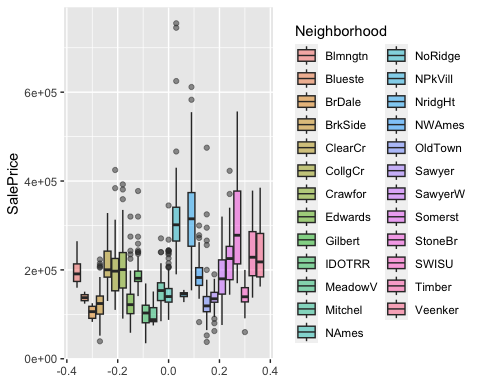
barplot(table(train2$Neighborhood), main = "Distribution of Neighborhood",xlab = "Number of Neighborhood",col = "skyblue")



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

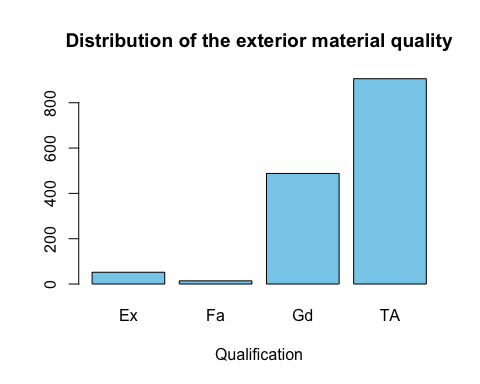
sum(is.na(train2$ExterQual)) #0

## [1] 0

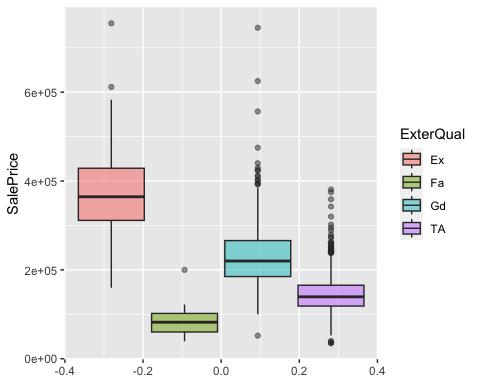
table(train2$ExterQual)

##   
## 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

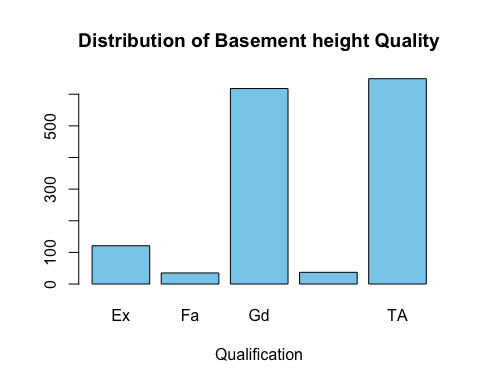
sum(is.na(train2$BsmtQual)) #0

## [1] 0

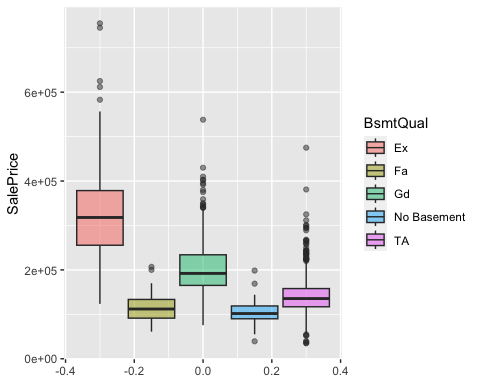
table(train2$BsmtQual)

##   
## 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

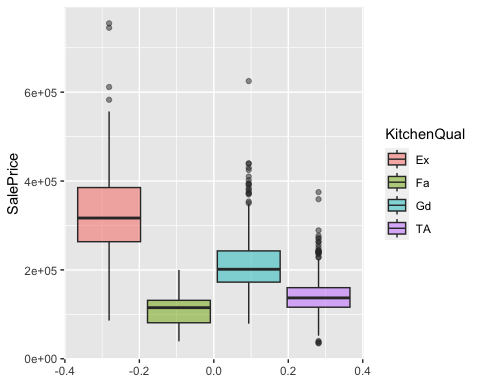
table(train2$KitchenQual)

##   
## 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

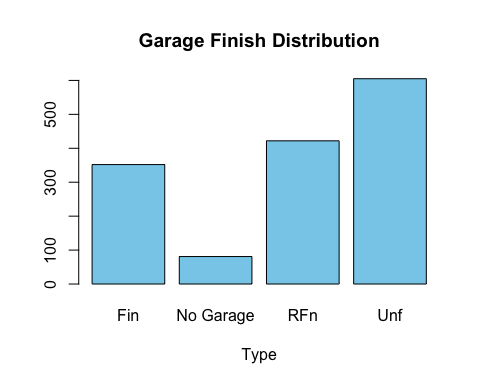
sum(is.na(train2$GarageFinish)) #0

## [1] 0

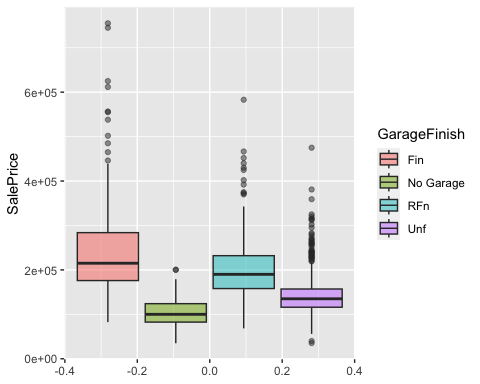
table(train2$GarageFinish)

##   
## 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

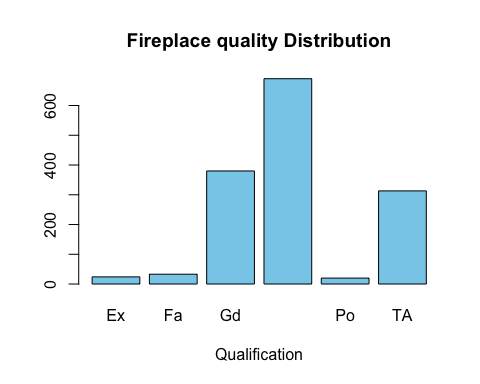
sum(is.na(train2$FireplaceQu)) #0

## [1] 0

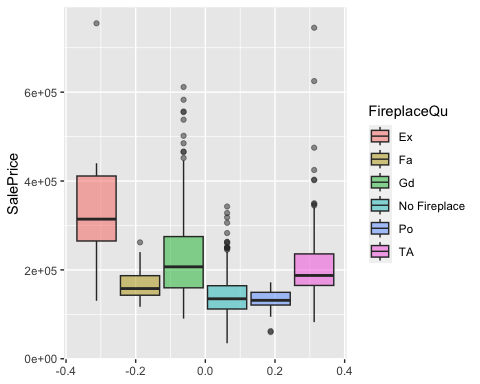
table(train2$FireplaceQu)

##   
## 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

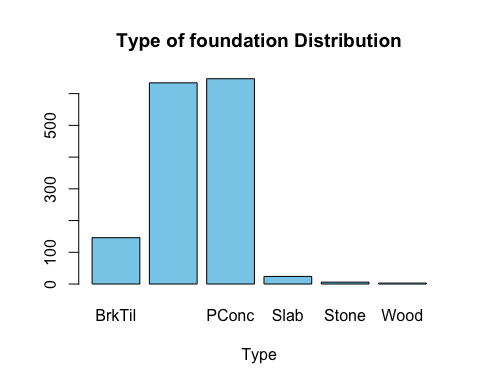
sum(is.na(train2$Foundation)) #0

## [1] 0

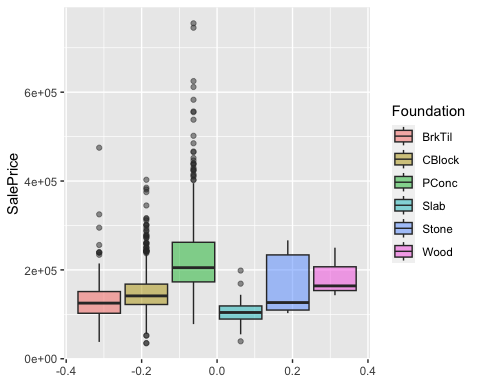
table(train2$Foundation)

##   
## 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

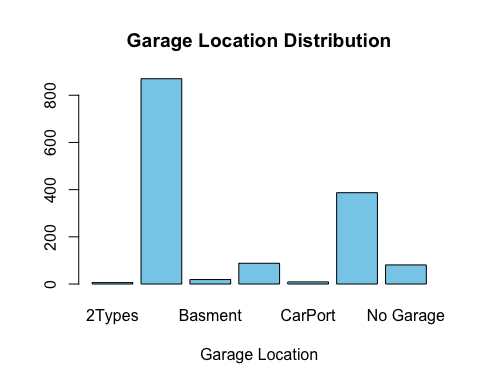
sum(is.na(train2$GarageType)) #0

## [1] 0

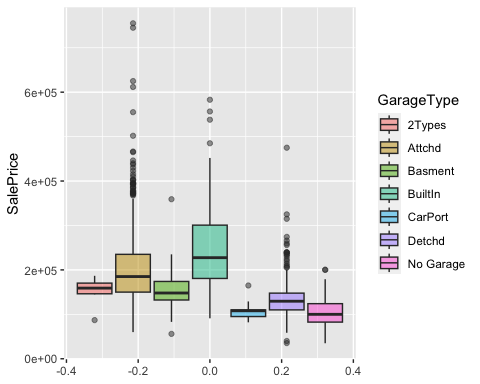
table(train2$GarageType)

##   
## 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

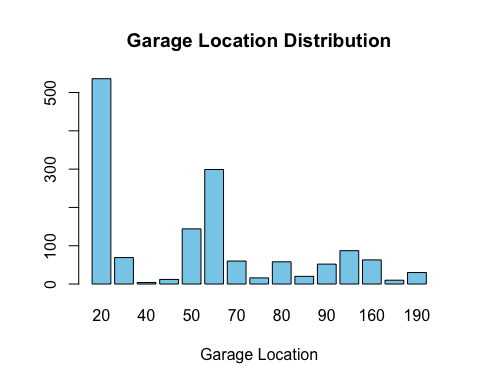
sum(is.na(train2$MSSubClass)) #0

## [1] 0

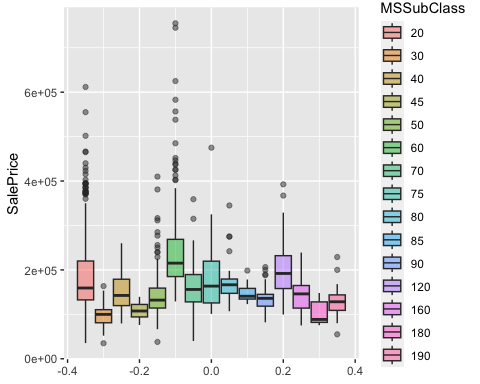
table(train2$MSSubClass) # +classes

##   
## 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

Checking which variables have NA’s we can see that LotFrontage, MasVnrArea and GarageYrBlt are the ones with missing values.

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

## [1] 259

sum(is.na(train2$MasVnrArea))

## [1] 10

sum(is.na(train2$GarageYrBlt))

## [1] 81

We decided to exclude the imputation of the GarageYrBlt feature because it has no sense to impute this variable, as it is a year.

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.

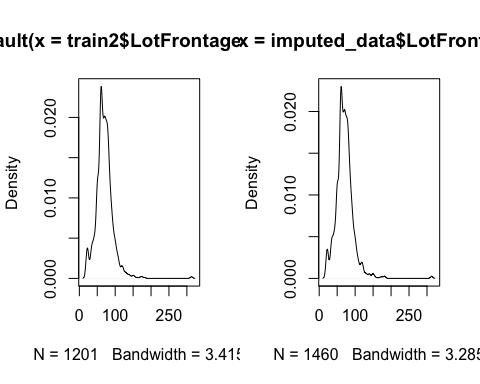
# LotFrontage validation  
summary(imputed\_data$LotFrontage)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 21.00 60.00 70.00 70.79 81.00 313.00

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

par(mfrow=c(1,2))  
plot(density(train2$LotFrontage,na.rm=TRUE))  
plot(density(imputed\_data$LotFrontage,na.rm=TRUE))



par(mfrow=c(1,1))

Checking the results we can conclude that the imputation doesn’t have a significant change in the density nor the summary.

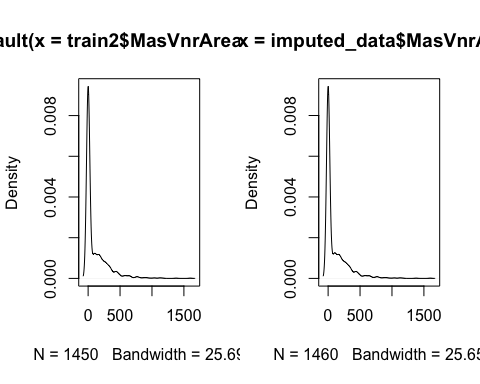
# MasVnrArea validation  
summary(imputed\_data$MasVnrArea)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0 0.0 0.0 103.3 164.0 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

par(mfrow=c(1,2))  
plot(density(train2$MasVnrArea,na.rm=TRUE))  
plot(density(imputed\_data$MasVnrArea,na.rm=TRUE))



par(mfrow=c(1,1))

# 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)

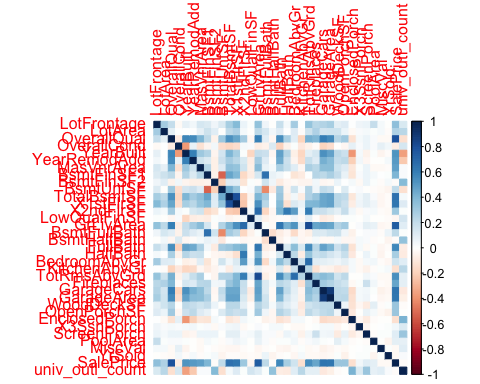
## [1] 10

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   
## 0.35051048 0.26760836 0.79609375 -0.07666732 0.52597552   
## YearRemodAdd MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF   
## 0.50833651 0.48779467 0.40868965 -0.01296088 0.21579373   
## TotalBsmtSF X1stFlrSF X2ndFlrSF LowQualFinSF GrLivArea   
## 0.65096994 0.63122613 0.31988123 -0.03308338 0.74067275   
## BsmtFullBath BsmtHalfBath FullBath HalfBath BedroomAbvGr   
## 0.22762274 -0.01687411 0.56184539 0.28530765 0.16741336   
## KitchenAbvGr TotRmsAbvGrd Fireplaces GarageCars GarageArea   
## -0.13877110 0.54284615 0.46954556 0.64249602 0.63038954   
## WoodDeckSF OpenPorchSF EnclosedPorch X3SsnPorch ScreenPorch   
## 0.32675949 0.32458454 -0.13663365 0.04469333 0.10961720   
## PoolArea MiscVal YrSold SalePrice univ\_outl\_count   
## 0.09827141 -0.02725834 -0.02770387 1.00000000 -0.01560585

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

## [1] "KitchenAbvGr"

## [1] -0.1387711

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

## [1] "OverallQual"

## [1] 0.7960937

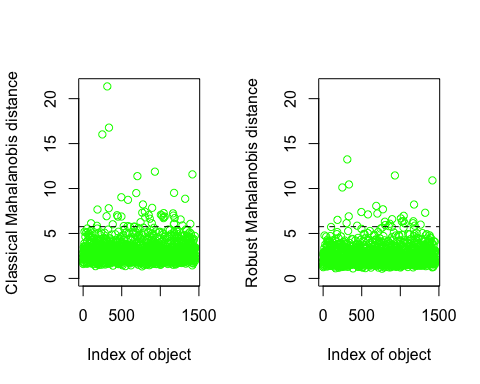
## Multivariate outliers

We 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. We are not taking off these outliers because the length is 44 and taking into account the dimensions of our dataset, 1460, it would be a reduction of the 3%. Also, we have performed other Moutliers in order to see different scenarios that we consider rare data.

# 1. All variable with similar high unique values  
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")



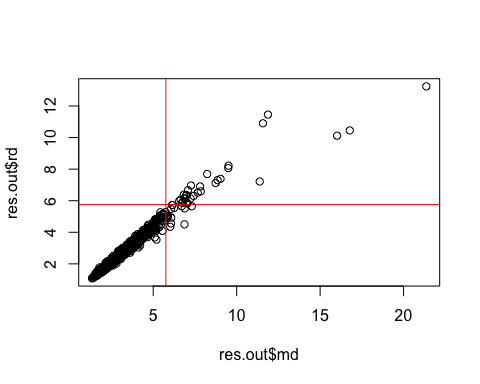
which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff))

## 186 250 305 314 333 336 441 496 584 692 707 779 804 808 855 869   
## 186 249 304 313 332 335 440 495 581 688 703 775 800 804 850 864   
## 899 935 1012 1025 1046 1049 1062 1170 1174 1183 1191 1269 1329 1424   
## 894 930 1007 1020 1041 1044 1057 1165 1169 1178 1186 1263 1322 1416

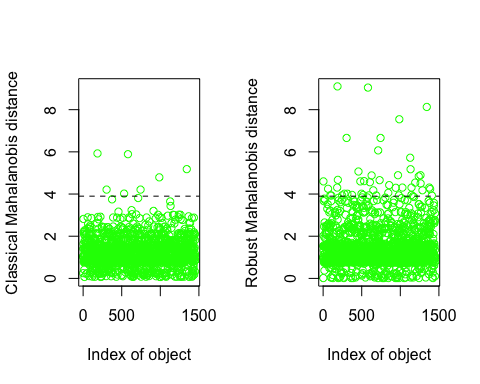
length(which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff))) #44

## [1] 30

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")



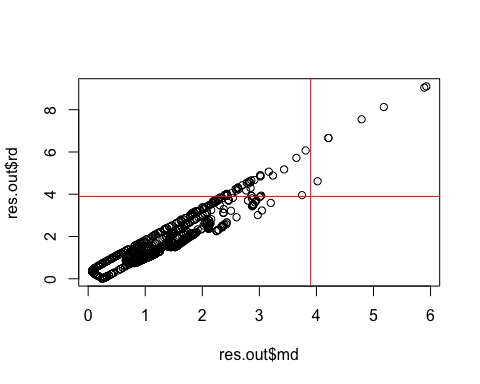
# take off m outliers  
#imputed\_data <- imputed\_data[-which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff)),]  
  
# 2. Generally the houses built latetly are the ones with most quality  
res.out <- Moutlier(imputed\_data[,c("YearBuilt","OverallQual")], quantile = 0.9995, col="green")



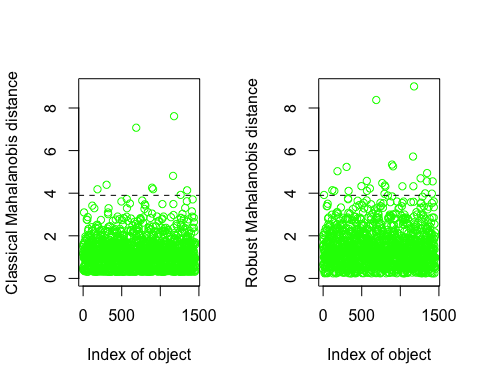
which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff))

## 186 305 534 584 748 992 1350   
## 186 304 531 581 744 987 1343

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")



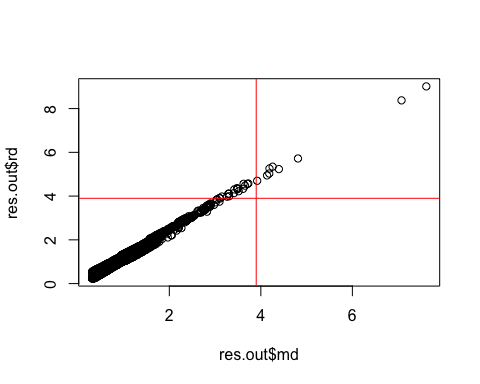
# 3. Generally the houses with more area are the ones with more rooms  
res.out <- Moutlier(imputed\_data[,c("GrLivArea","TotRmsAbvGrd")], quantile = 0.9995, col="green")



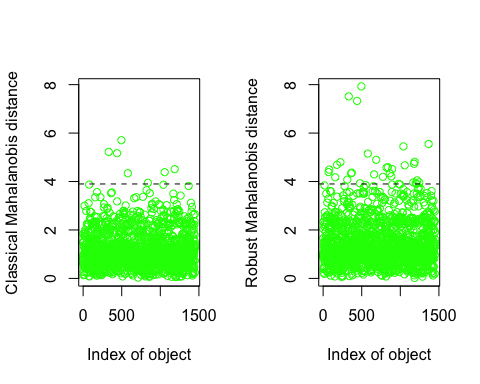
which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff))

## 186 305 692 898 911 1170 1183 1269 1354   
## 186 304 688 893 906 1165 1178 1263 1347

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")



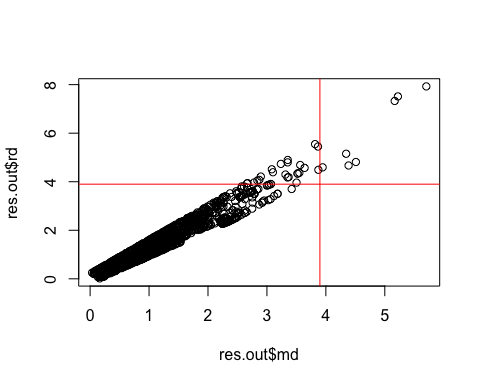
# 4. Generally the houses with more total area are the ones with more garage area also  
res.out <- Moutlier(imputed\_data[,c("GarageArea","TotalBsmtSF")], quantile = 0.9995, col="green")



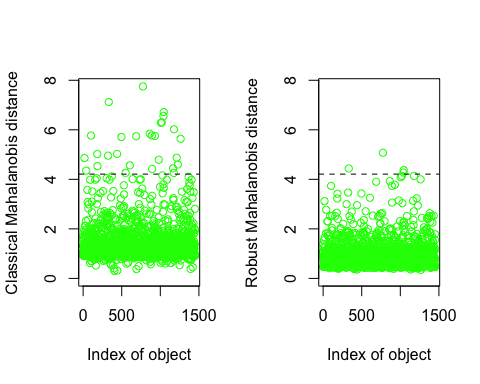
which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff))

## 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")



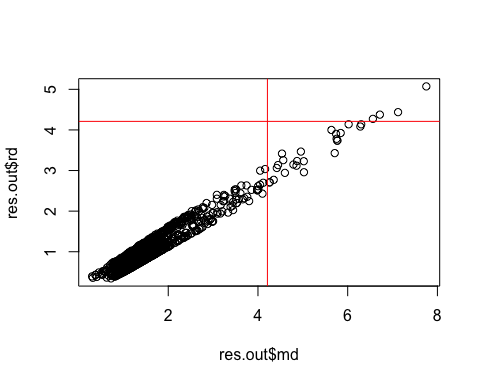
# 5. Moutliers about total areas zones  
res.out <- Moutlier(imputed\_data[,c("X1stFlrSF","TotalBsmtSF","GrLivArea")], quantile = 0.9995, col="green") #GarageArea, si se lo sumo hay menos outliers



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

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

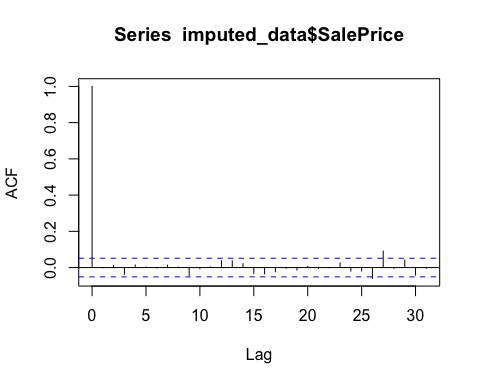
par(mfrow=c(1,1))  
plot( res.out$md, res.out$rd )  
abline(h=res.out$cutoff, col="red")  
abline(v=res.out$cutoff, col="red")



# Profiling

As we have declared before in the target variable exploration, the distribution is not normal. To test for autocorrelation the acf() function is used.

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 = 0.86922, p-value < 2.2e-16

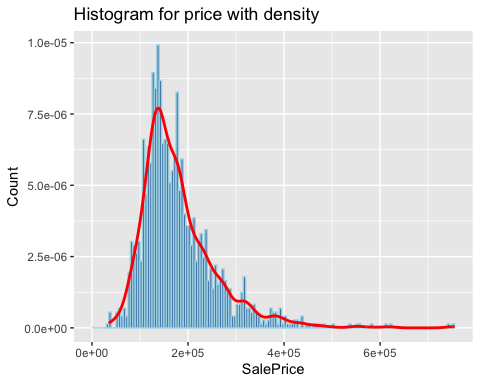
ks.test(imputed\_data$SalePrice, 'pnorm', mean(imputed\_data$SalePrice), sd(imputed\_data$SalePrice))

## 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 = 0.12376, p-value < 2.2e-16  
## alternative hypothesis: two-sided

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 0.79609375 1.398670e-318  
## GrLivArea 0.74067275 1.025988e-252  
## TotalBsmtSF 0.65096994 8.783474e-176  
## GarageCars 0.64249602 7.623680e-170  
## X1stFlrSF 0.63122613 3.121909e-162  
## GarageArea 0.63038954 1.114177e-161  
## FullBath 0.56184539 1.375148e-121  
## TotRmsAbvGrd 0.54284615 4.622921e-112  
## YearBuilt 0.52597552 4.236239e-104  
## YearRemodAdd 0.50833651 3.052623e-96  
## MasVnrArea 0.48779467 1.182039e-87  
## Fireplaces 0.46954556 1.682802e-80  
## BsmtFinSF1 0.40868965 1.464343e-59  
## LotFrontage 0.35051048 3.163720e-43  
## WoodDeckSF 0.32675949 1.787547e-37  
## OpenPorchSF 0.32458454 5.676640e-37  
## X2ndFlrSF 0.31988123 6.688459e-36  
## HalfBath 0.28530765 1.346076e-28  
## LotArea 0.26760836 3.111931e-25  
## BsmtFullBath 0.22762274 1.611002e-18  
## BsmtUnfSF 0.21579373 9.260246e-17  
## BedroomAbvGr 0.16741336 1.370234e-10  
## ScreenPorch 0.10961720 2.839149e-05  
## PoolArea 0.09827141 1.764824e-04  
## OverallCond -0.07666732 3.464488e-03  
## EnclosedPorch -0.13663365 1.727619e-07  
## KitchenAbvGr -0.13877110 1.102034e-07

#done before for selecting  
res.con$quali

## R2 p.value  
## Neighborhood 0.54755148 1.595761e-225  
## ExterQual 0.48847936 3.501182e-210  
## BsmtQual 0.47147373 1.488792e-198  
## KitchenQual 0.46306298 6.028665e-195  
## GarageFinish 0.31013396 3.123779e-116  
## FireplaceQu 0.29575431 1.902525e-107  
## Foundation 0.25784442 4.504809e-91  
## GarageType 0.25257359 7.585307e-88  
## MSSubClass 0.24826775 4.225305e-79  
## BsmtFinType1 0.21246055 1.362078e-71  
## OpenPorch\_binary 0.17113708 4.009012e-61  
## HasPorch\_binary 0.09173377 3.479772e-32  
## EnclosedPorch\_binary 0.03445322 1.012072e-12  
## f.LotArea 0.03035151 2.004842e-10  
## secondfloor 0.01860096 1.819444e-07  
## Pool\_binary 0.01146507 4.344490e-05

# 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. 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.

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] 0.8341302

## 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] 0.8080929

## 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] 0.7947344

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

## Build different models with their metrics

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 forward stepwise step() and see which are the variables that will create a model with higher accuracy.

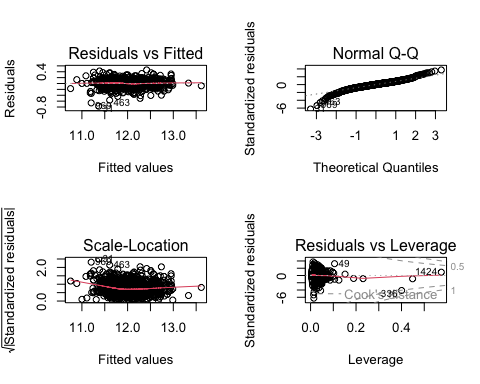
mod.fow <- stats::step(lm(SalePrice~. , data = train\_new), trace = FALSE,  
 direction = "forward")  
summary(mod.fow)

##   
## 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 + f.LotArea + secondfloor +   
## OpenPorch\_binary + EnclosedPorch\_binary + HasPorch\_binary +   
## Pool\_binary, data = train\_new)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -91019 -10510 782 10365 167658   
##   
## Coefficients: (4 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.162e+06 1.237e+06 1.748 0.080812 .   
## Id 1.360e+00 1.880e+00 0.724 0.469471   
## LotFrontage 1.006e+02 4.494e+01 2.238 0.025450 \*   
## LotArea 5.694e-01 1.439e-01 3.958 8.16e-05 \*\*\*  
## OverallQual 8.622e+03 1.183e+03 7.286 6.96e-13 \*\*\*  
## OverallCond 6.578e+03 9.320e+02 7.058 3.37e-12 \*\*\*  
## YearBuilt 4.481e+02 9.711e+01 4.614 4.52e-06 \*\*\*  
## YearRemodAdd 9.526e+01 6.146e+01 1.550 0.121486   
## MasVnrArea 2.079e+01 5.657e+00 3.675 0.000252 \*\*\*  
## BsmtFinSF1 4.388e+01 5.885e+00 7.457 2.07e-13 \*\*\*  
## BsmtFinSF2 3.635e+01 8.411e+00 4.321 1.73e-05 \*\*\*  
## BsmtUnfSF 2.812e+01 5.396e+00 5.212 2.32e-07 \*\*\*  
## TotalBsmtSF NA NA NA NA   
## X1stFlrSF 4.220e+01 6.230e+00 6.773 2.27e-11 \*\*\*  
## X2ndFlrSF 7.740e+01 7.210e+00 10.735 < 2e-16 \*\*\*  
## LowQualFinSF 1.493e+01 2.000e+01 0.746 0.455608   
## GrLivArea NA NA NA NA   
## BsmtFullBath 1.558e+03 2.364e+03 0.659 0.510095   
## BsmtHalfBath 5.875e+03 4.411e+03 1.332 0.183296   
## FullBath 5.785e+03 2.645e+03 2.188 0.028949 \*   
## HalfBath 3.443e+03 2.493e+03 1.381 0.167523   
## BedroomAbvGr -5.582e+03 1.650e+03 -3.382 0.000750 \*\*\*  
## KitchenAbvGr -7.998e+03 6.666e+03 -1.200 0.230531   
## TotRmsAbvGrd 3.975e+02 1.167e+03 0.341 0.733469   
## Fireplaces 5.626e+03 3.044e+03 1.848 0.064893 .   
## GarageCars 6.623e+03 2.593e+03 2.554 0.010814 \*   
## GarageArea 6.844e+00 8.905e+00 0.769 0.442389   
## WoodDeckSF 1.812e+01 6.990e+00 2.592 0.009692 \*\*   
## OpenPorchSF 3.074e+01 1.801e+01 1.707 0.088129 .   
## EnclosedPorch -1.938e+01 3.081e+01 -0.629 0.529614   
## X3SsnPorch 4.247e+01 2.482e+01 1.711 0.087419 .   
## ScreenPorch 6.480e+01 1.900e+01 3.410 0.000679 \*\*\*  
## PoolArea 1.978e+02 4.162e+02 0.475 0.634811   
## MiscVal 4.738e+00 6.176e+00 0.767 0.443189   
## YrSold -1.590e+03 6.081e+02 -2.615 0.009071 \*\*   
## NeighborhoodBlueste -1.030e+03 2.648e+04 -0.039 0.968982   
## NeighborhoodBrDale -1.333e+04 1.272e+04 -1.048 0.294926   
## NeighborhoodBrkSide -1.381e+04 1.069e+04 -1.291 0.196871   
## NeighborhoodClearCr -3.247e+04 1.109e+04 -2.928 0.003495 \*\*   
## NeighborhoodCollgCr -2.340e+04 9.254e+03 -2.528 0.011629 \*   
## NeighborhoodCrawfor 3.367e+03 1.033e+04 0.326 0.744528   
## NeighborhoodEdwards -2.748e+04 9.917e+03 -2.771 0.005698 \*\*   
## NeighborhoodGilbert -2.443e+04 9.550e+03 -2.558 0.010676 \*   
## NeighborhoodIDOTRR -2.556e+04 1.134e+04 -2.255 0.024398 \*   
## NeighborhoodMeadowV -7.704e+03 1.286e+04 -0.599 0.549337   
## NeighborhoodMitchel -3.499e+04 1.021e+04 -3.428 0.000636 \*\*\*  
## NeighborhoodNAmes -2.483e+04 9.639e+03 -2.576 0.010156 \*   
## NeighborhoodNoRidge 3.099e+03 1.049e+04 0.296 0.767641   
## NeighborhoodNPkVill 9.708e+02 1.321e+04 0.073 0.941443   
## NeighborhoodNridgHt -6.215e+03 9.557e+03 -0.650 0.515630   
## NeighborhoodNWAmes -3.221e+04 9.870e+03 -3.264 0.001141 \*\*   
## NeighborhoodOldTown -2.904e+04 1.028e+04 -2.824 0.004850 \*\*   
## NeighborhoodSawyer -2.480e+04 9.985e+03 -2.484 0.013187 \*   
## NeighborhoodSawyerW -2.248e+04 9.740e+03 -2.308 0.021220 \*   
## NeighborhoodSomerst -8.776e+03 9.629e+03 -0.911 0.362347   
## NeighborhoodStoneBr 2.375e+04 1.033e+04 2.298 0.021809 \*   
## NeighborhoodSWISU -2.496e+04 1.162e+04 -2.148 0.031962 \*   
## NeighborhoodTimber -2.417e+04 1.010e+04 -2.393 0.016937 \*   
## NeighborhoodVeenker 2.250e+03 1.306e+04 0.172 0.863234   
## ExterQualFa -1.299e+04 1.117e+04 -1.163 0.245299   
## ExterQualGd -2.119e+04 5.439e+03 -3.895 0.000105 \*\*\*  
## ExterQualTA -2.160e+04 6.070e+03 -3.559 0.000392 \*\*\*  
## BsmtQualFa -2.754e+04 7.301e+03 -3.773 0.000172 \*\*\*  
## BsmtQualGd -2.850e+04 4.220e+03 -6.753 2.59e-11 \*\*\*  
## BsmtQualNo Basement 1.075e+03 1.197e+04 0.090 0.928441   
## BsmtQualTA -2.583e+04 5.137e+03 -5.028 5.99e-07 \*\*\*  
## KitchenQualFa -1.768e+04 7.541e+03 -2.345 0.019262 \*   
## KitchenQualGd -1.417e+04 4.095e+03 -3.460 0.000566 \*\*\*  
## KitchenQualTA -1.568e+04 4.615e+03 -3.397 0.000711 \*\*\*  
## GarageFinishNo Garage 2.700e+04 1.434e+04 1.883 0.060073 .   
## GarageFinishRFn -2.368e+03 2.442e+03 -0.970 0.332457   
## GarageFinishUnf 1.178e+02 2.962e+03 0.040 0.968297   
## FireplaceQuFa -1.690e+04 8.161e+03 -2.071 0.038647 \*   
## FireplaceQuGd -1.071e+04 6.323e+03 -1.694 0.090582 .   
## FireplaceQuNo Fireplace -4.770e+03 7.669e+03 -0.622 0.534098   
## FireplaceQuPo -9.949e+03 9.073e+03 -1.097 0.273122   
## FireplaceQuTA -1.336e+04 6.578e+03 -2.031 0.042587 \*   
## FoundationCBlock 4.179e+02 3.734e+03 0.112 0.910904   
## FoundationPConc 3.641e+03 4.245e+03 0.858 0.391259   
## FoundationSlab -1.568e+03 1.135e+04 -0.138 0.890194   
## FoundationStone 1.736e+04 1.484e+04 1.170 0.242287   
## FoundationWood -3.113e+04 1.578e+04 -1.973 0.048854 \*   
## GarageTypeAttchd 1.456e+04 1.314e+04 1.109 0.267911   
## GarageTypeBasment 2.421e+04 1.515e+04 1.599 0.110261   
## GarageTypeBuiltIn 2.053e+04 1.373e+04 1.495 0.135228   
## GarageTypeCarPort -4.589e+03 1.609e+04 -0.285 0.775624   
## GarageTypeDetchd 1.650e+04 1.313e+04 1.256 0.209314   
## GarageTypeNo Garage NA NA NA NA   
## MSSubClass30 3.616e+03 5.579e+03 0.648 0.517117   
## MSSubClass40 1.714e+04 2.549e+04 0.672 0.501447   
## MSSubClass45 3.823e+02 9.786e+03 0.039 0.968842   
## MSSubClass50 1.194e+04 6.819e+03 1.752 0.080197 .   
## MSSubClass60 9.758e+01 6.235e+03 0.016 0.987516   
## MSSubClass70 4.456e+03 7.917e+03 0.563 0.573665   
## MSSubClass75 1.356e+04 1.131e+04 1.198 0.231089   
## MSSubClass80 3.211e+03 4.841e+03 0.663 0.507262   
## MSSubClass85 3.042e+03 7.270e+03 0.418 0.675708   
## MSSubClass90 -1.113e+04 6.820e+03 -1.632 0.103005   
## MSSubClass120 -2.172e+04 4.818e+03 -4.509 7.36e-06 \*\*\*  
## MSSubClass160 -2.375e+04 8.420e+03 -2.820 0.004901 \*\*   
## MSSubClass180 -2.111e+04 1.244e+04 -1.697 0.090060 .   
## MSSubClass190 1.908e+02 7.618e+03 0.025 0.980020   
## BsmtFinType1BLQ 3.793e+03 3.233e+03 1.173 0.240966   
## BsmtFinType1GLQ 3.303e+03 3.012e+03 1.097 0.273089   
## BsmtFinType1LwQ -4.445e+03 4.438e+03 -1.001 0.316856   
## BsmtFinType1No Basement NA NA NA NA   
## BsmtFinType1Rec -3.448e+03 3.426e+03 -1.007 0.314414   
## BsmtFinType1Unf -3.147e+02 3.350e+03 -0.094 0.925178   
## univ\_outl\_count -4.042e+03 2.563e+03 -1.577 0.115185   
## f.LotArea.L 9.337e+03 3.874e+03 2.410 0.016158 \*   
## secondfloor1 -2.571e+04 7.232e+03 -3.556 0.000396 \*\*\*  
## OpenPorch\_binary1 1.273e+03 5.199e+03 0.245 0.806548   
## EnclosedPorch\_binary1 1.308e+04 7.275e+03 1.798 0.072457 .   
## HasPorch\_binary1 -3.978e+03 5.299e+03 -0.751 0.453021   
## Pool\_binary1 -1.021e+05 2.871e+05 -0.356 0.722190   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 23540 on 905 degrees of freedom  
## Multiple R-squared: 0.9198, Adjusted R-squared: 0.9101   
## F-statistic: 94.36 on 110 and 905 DF, p-value: < 2.2e-16

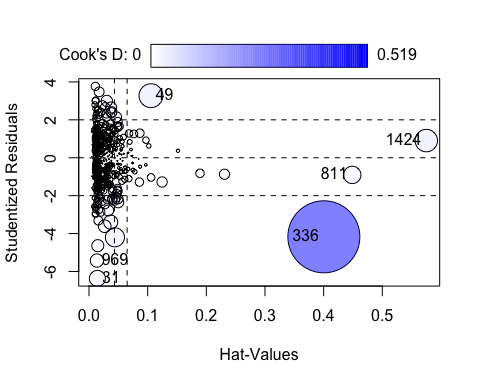
# Model:   
  
n00<- lm(log(SalePrice) ~ LotArea + OverallQual + OverallCond + YearBuilt +  
 MasVnrArea + BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF +  
 X1stFlrSF + X2ndFlrSF +  
 BsmtFullBath + FullBath + HalfBath + BedroomAbvGr +  
 KitchenAbvGr + Fireplaces + GarageCars +  
 OpenPorchSF + X3SsnPorch +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

## [1] 0.9042693

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

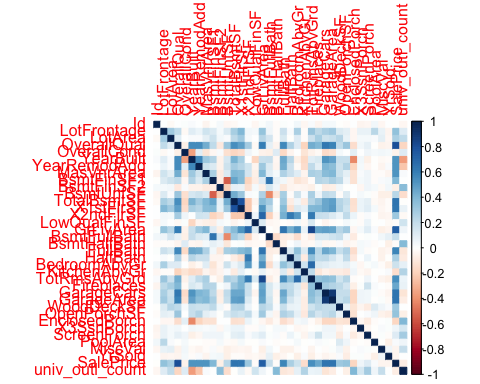


par(mfrow=c(1,1))  
influencePlot(n00)



## StudRes Hat CookD  
## 811 -0.9049556 0.44874356 0.03030788  
## 31 -6.3655791 0.01426870 0.02564167  
## 49 3.2745216 0.10554476 0.05695395  
## 336 -4.1700516 0.40034283 0.51914280  
## 969 -5.4233711 0.01345858 0.01773209  
## 1424 0.9049556 0.57499801 0.05037167

corrplot(descrCor, method = 'color')



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

## [1] 10.44598

vif(n00) # there are not

## LotArea OverallQual OverallCond YearBuilt MasVnrArea BsmtFinSF1   
## 1.203015 3.298338 1.252738 2.835042 1.429854 5.357236   
## BsmtFinSF2 BsmtUnfSF X1stFlrSF X2ndFlrSF BsmtFullBath FullBath   
## 1.532334 4.334549 4.482202 4.576288 1.980807 2.892160   
## HalfBath BedroomAbvGr KitchenAbvGr Fireplaces GarageCars OpenPorchSF   
## 2.246362 1.670131 1.286969 1.529942 1.967591 1.188721   
## X3SsnPorch ScreenPorch PoolArea   
## 1.014926 1.063377 1.019178

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)

## Plot to show which transformations we shall apply

# 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|)   
## -0.3036 -6.7678 1.307e-11 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## iterations = 5

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|)   
## 0.50049 -2.8813 0.00396 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## iterations = 3

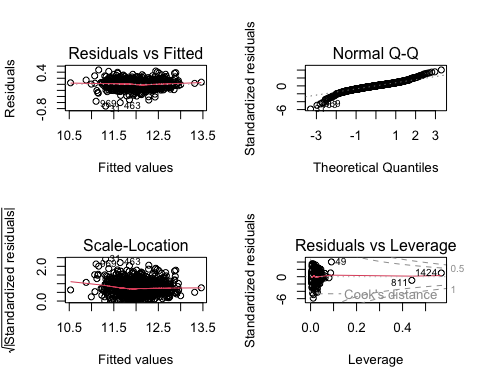
v1<- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + BsmtFinSF2+ log(X1stFlrSF)+ X2ndFlrSF +  
 BsmtFullBath + KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea, data = train\_new)  
  
  
# X2stFlrSF  
  
boxTidwell(log(SalePrice) ~ I(X2ndFlrSF+0.01),~ 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|)  
## 1.1024 1.0891 0.2761  
##   
## iterations = 2

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

## [1] 0.904978

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)

## Select the best model

Therefore we have 2 potential models. We have been looking on the accuracy of the model, currently on 87% (will depend on the seed), but we need to check if there is overfitting using the test sample. Looking at these results, seems that the model has no overfitting. No significant difference on the measures of the model between the train sample and the test.

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 23661.06  
## 2 Model with test\_new sample 0.90 25469.78

## Add categorical variables to the model

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] 0.9207975

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 1001 15.245   
## 2 975 12.707 26 2.538 7.4901 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(v1,b0)

## df AIC  
## v1 16 -1351.285  
## b0 42 -1484.300

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

## [1] 12.62587

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

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.868388 1 1.366890  
## log(LotArea) 2.901651 1 1.703423  
## OverallQual 3.646170 1 1.909495  
## OverallCond 1.345390 1 1.159910  
## YearBuilt 6.069015 1 2.463537  
## BsmtFinSF1 2.187969 1 1.479179  
## log(X1stFlrSF) 4.524469 1 2.127080  
## X2ndFlrSF 4.607762 1 2.146570  
## BsmtFullBath 1.898844 1 1.377985  
## FullBath 2.595181 1 1.610957  
## TotRmsAbvGrd 3.970172 1 1.992529  
## KitchenAbvGr 1.391748 1 1.179724  
## Fireplaces 1.684821 1 1.298007  
## GarageCars 2.133265 1 1.460570  
## ScreenPorch 1.093399 1 1.045657  
## PoolArea 1.110567 1 1.053835  
## Neighborhood 43.855980 24 1.081954

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] 0.9091318

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 1001 16.197   
## 2 996 14.579 5 1.6181 22.11 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(n0,b1)

## df AIC  
## n0 16 -1289.761  
## b1 21 -1386.699

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

## [1] 11.00495

vif(b1)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.795392 1 1.339923  
## log(LotArea) 1.893757 1 1.376139  
## OverallQual 3.558936 1 1.886514  
## OverallCond 1.288562 1 1.135149  
## YearBuilt 2.577264 1 1.605386  
## BsmtFinSF1 2.064791 1 1.436938  
## log(X1stFlrSF) 4.043197 1 2.010770  
## X2ndFlrSF 4.045079 1 2.011238  
## BsmtFullBath 1.825702 1 1.351185  
## FullBath 2.353893 1 1.534240  
## TotRmsAbvGrd 3.772948 1 1.942408  
## KitchenAbvGr 1.318914 1 1.148440  
## Fireplaces 1.490503 1 1.220862  
## GarageCars 1.987945 1 1.409945  
## ScreenPorch 1.062434 1 1.030745  
## PoolArea 1.087668 1 1.042913  
## ExterQual 2.932254 3 1.196374

# 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] 0.9164236

anova(b1,b11) # the model is better with interactions but for explainability reasons we will not consider the 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)   
## 1 996 14.579   
## 2 958 13.409 38 1.1699 2.1996 5.044e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b1,b11)

## df AIC  
## b1 21 -1386.699  
## b11 59 -1395.687

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] 0.9137113

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)   
## 1 996 14.579   
## 2 992 13.844 4 0.73472 13.162 1.887e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b1,b2)

## df AIC  
## b1 21 -1386.699  
## b2 25 -1431.238

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

## [1] 11.58901

vif(b2)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.804182 1 1.343198  
## log(LotArea) 1.904537 1 1.380050  
## OverallQual 3.826496 1 1.956143  
## OverallCond 1.309493 1 1.144331  
## YearBuilt 3.301283 1 1.816943  
## BsmtFinSF1 2.119057 1 1.455698  
## log(X1stFlrSF) 4.048725 1 2.012144  
## X2ndFlrSF 4.116511 1 2.028919  
## BsmtFullBath 1.839948 1 1.356447  
## FullBath 2.463106 1 1.569428  
## TotRmsAbvGrd 3.918411 1 1.979498  
## KitchenAbvGr 1.419645 1 1.191488  
## Fireplaces 1.494777 1 1.222611  
## GarageCars 2.018785 1 1.420839  
## ScreenPorch 1.065690 1 1.032323  
## PoolArea 1.088436 1 1.043281  
## ExterQual 3.813635 3 1.249942  
## BsmtQual 5.693745 4 1.242867

# 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] 0.9207653

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)   
## 1 992 13.844   
## 2 929 12.712 63 1.1317 1.3128 0.0559 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b2,b21)

## df AIC  
## b2 25 -1431.238  
## b21 88 -1391.886

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 13.409   
## 2 929 12.712 29 0.69655 1.7553 0.008439 \*\*  
## ---  
## 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] 0.9149272

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)   
## 1 992 13.844   
## 2 989 13.649 3 0.19507 4.7116 0.002849 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b2,b3)

## df AIC  
## b2 25 -1431.238  
## b3 28 -1439.656

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

## [1] 11.75464

vif(b3)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.817057 1 1.347983  
## log(LotArea) 1.907361 1 1.381072  
## OverallQual 3.910892 1 1.977597  
## OverallCond 1.378000 1 1.173882  
## YearBuilt 3.381388 1 1.838855  
## BsmtFinSF1 2.127912 1 1.458736  
## log(X1stFlrSF) 4.098912 1 2.024577  
## X2ndFlrSF 4.152745 1 2.037828  
## BsmtFullBath 1.848176 1 1.359476  
## FullBath 2.499374 1 1.580941  
## TotRmsAbvGrd 3.941598 1 1.985346  
## KitchenAbvGr 1.439760 1 1.199900  
## Fireplaces 1.496105 1 1.223154  
## GarageCars 2.059253 1 1.435010  
## ScreenPorch 1.069102 1 1.033974  
## PoolArea 1.092751 1 1.045347  
## ExterQual 5.800572 3 1.340433  
## BsmtQual 6.290120 4 1.258440  
## KitchenQual 4.883735 3 1.302543

#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] 0.9238925

anova(b3,b31) # better model with interactions but we will not consider them for explainability pourposes

## 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)   
## 1 989 13.649   
## 2 931 12.210 58 1.4384 1.8909 0.0001018 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b3,b31)

## df AIC  
## b3 28 -1439.656  
## b31 86 -1436.798

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] 0.9151954

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)  
## 1 989 13.649   
## 2 986 13.606 3 0.043026 1.0394 0.3743

AIC(b3,b4)

## df AIC  
## b3 28 -1439.656  
## b4 31 -1436.864

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

## [1] 11.79181

vif(b4)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.833943 1 1.354231  
## log(LotArea) 1.910960 1 1.382375  
## OverallQual 3.924208 1 1.980961  
## OverallCond 1.381484 1 1.175365  
## YearBuilt 3.612547 1 1.900670  
## BsmtFinSF1 2.138330 1 1.462303  
## log(X1stFlrSF) 4.112557 1 2.027944  
## X2ndFlrSF 4.161175 1 2.039896  
## BsmtFullBath 1.853011 1 1.361253  
## FullBath 2.568837 1 1.602759  
## TotRmsAbvGrd 3.978225 1 1.994549  
## KitchenAbvGr 1.461015 1 1.208725  
## Fireplaces 1.524441 1 1.234683  
## GarageCars 3.025833 1 1.739492  
## ScreenPorch 1.071712 1 1.035235  
## PoolArea 1.094295 1 1.046085  
## ExterQual 5.996619 3 1.347880  
## BsmtQual 6.603205 4 1.266104  
## KitchenQual 4.976765 3 1.306646  
## GarageFinish 3.821136 3 1.250351

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] 0.9157543

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)   
## 1 989 13.649   
## 2 984 13.516 5 0.1327 1.9322 0.08647 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b3,b5)

## df AIC  
## b3 28 -1439.656  
## b5 33 -1439.582

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

## [1] 11.87004

vif(b5)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.825405 1 1.351076  
## log(LotArea) 1.920993 1 1.385999  
## OverallQual 3.967984 1 1.991980  
## OverallCond 1.381023 1 1.175170  
## YearBuilt 3.443870 1 1.855767  
## BsmtFinSF1 2.159883 1 1.469654  
## log(X1stFlrSF) 4.143578 1 2.035578  
## X2ndFlrSF 4.245403 1 2.060438  
## BsmtFullBath 1.860438 1 1.363979  
## FullBath 2.517866 1 1.586778  
## TotRmsAbvGrd 3.998023 1 1.999506  
## KitchenAbvGr 1.455081 1 1.206268  
## Fireplaces 5.984544 1 2.446333  
## GarageCars 2.065290 1 1.437112  
## ScreenPorch 1.078810 1 1.038658  
## PoolArea 1.111790 1 1.054414  
## ExterQual 5.883696 3 1.343616  
## BsmtQual 6.670758 4 1.267716  
## KitchenQual 5.073848 3 1.310860  
## FireplaceQu 8.888261 5 1.244176

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] 0.9165535

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)   
## 1 989 13.649   
## 2 984 13.388 5 0.26092 3.8355 0.001904 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b3,b6)

## df AIC  
## b3 28 -1439.656  
## b6 33 -1449.266

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

## [1] 11.98372

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

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.912456 1 1.382916  
## log(LotArea) 1.917078 1 1.384586  
## OverallQual 4.028412 1 2.007090  
## OverallCond 1.433233 1 1.197177  
## YearBuilt 5.499491 1 2.345099  
## BsmtFinSF1 2.163740 1 1.470966  
## log(X1stFlrSF) 4.162534 1 2.040229  
## X2ndFlrSF 4.219426 1 2.054124  
## BsmtFullBath 1.853254 1 1.361343  
## FullBath 2.533465 1 1.591686  
## TotRmsAbvGrd 3.965050 1 1.991243  
## KitchenAbvGr 1.441761 1 1.200733  
## Fireplaces 1.526900 1 1.235678  
## GarageCars 2.103998 1 1.450517  
## ScreenPorch 1.072583 1 1.035656  
## PoolArea 1.097291 1 1.047516  
## ExterQual 6.095982 3 1.351576  
## BsmtQual 19.251031 4 1.447294  
## KitchenQual 5.154560 3 1.314312  
## Foundation 17.614725 5 1.332256

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] 0.9160752

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)   
## 1 989 13.649   
## 2 983 13.465 6 0.18418 2.241 0.03735 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b3,b7)

## df AIC  
## b3 28 -1439.656  
## b7 34 -1441.459

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

## [1] 11.91542

vif(b7)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.851006 1 1.360517  
## log(LotArea) 1.939535 1 1.392672  
## OverallQual 3.959938 1 1.989959  
## OverallCond 1.389501 1 1.178771  
## YearBuilt 3.982877 1 1.995715  
## BsmtFinSF1 2.145075 1 1.464607  
## log(X1stFlrSF) 4.203464 1 2.050235  
## X2ndFlrSF 4.365694 1 2.089424  
## BsmtFullBath 1.852967 1 1.361237  
## FullBath 2.543950 1 1.594976  
## TotRmsAbvGrd 3.978506 1 1.994619  
## KitchenAbvGr 1.506834 1 1.227532  
## Fireplaces 1.530406 1 1.237096  
## GarageCars 3.199960 1 1.788843  
## ScreenPorch 1.074708 1 1.036681  
## PoolArea 1.094024 1 1.045956  
## ExterQual 6.016196 3 1.348612  
## BsmtQual 6.656814 4 1.267384  
## KitchenQual 5.091216 3 1.311607  
## GarageType 4.805752 6 1.139760

# 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] 0.9273753

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)   
## 1 983 13.465   
## 2 919 11.652 64 1.813 2.2343 2.871e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b7,b71)

## df AIC  
## b7 34 -1441.459  
## b71 98 -1460.390

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

## [1] 13.76942

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] 0.9176299

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)  
## 1 983 13.465   
## 2 969 13.215 14 0.24944 1.3065 0.1964

AIC(b7,b8)

## df AIC  
## b7 34 -1441.459  
## b8 48 -1432.458

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

## [1] 12.14033

vif(b8) # highly correlated

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.983909 1 1.408513  
## log(LotArea) 2.813599 1 1.677379  
## OverallQual 4.205270 1 2.050675  
## OverallCond 1.450044 1 1.204178  
## YearBuilt 8.531955 1 2.920951  
## BsmtFinSF1 2.189428 1 1.479672  
## log(X1stFlrSF) 4.857300 1 2.203928  
## X2ndFlrSF 9.266837 1 3.044148  
## BsmtFullBath 1.925935 1 1.387781  
## FullBath 2.600312 1 1.612548  
## TotRmsAbvGrd 4.223516 1 2.055119  
## KitchenAbvGr 2.962923 1 1.721314  
## Fireplaces 1.606857 1 1.267619  
## GarageCars 3.278739 1 1.810729  
## ScreenPorch 1.085780 1 1.042008  
## PoolArea 1.112646 1 1.054820  
## ExterQual 6.406185 3 1.362804  
## BsmtQual 8.276342 4 1.302356  
## KitchenQual 5.444139 3 1.326340  
## GarageType 6.948443 6 1.175323  
## MSSubClass 408.828805 14 1.239565

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] 0.9164602

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)  
## 1 983 13.465   
## 2 978 13.403 5 0.061777 0.9016 0.4793

AIC(b7,b9)

## df AIC  
## b7 34 -1441.459  
## b9 39 -1436.131

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

## [1] 11.97034

#f.LotArea  
b10<- 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+f.LotArea  
 , data = train\_new)  
  
#summary(b10)  
summary(b10)$r.squared

## [1] 0.9160844

anova(b7,b10) # 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 + f.LotArea  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 983 13.465   
## 2 982 13.463 1 0.0014756 0.1076 0.7429

AIC(b7,b10)

## df AIC  
## b7 34 -1441.459  
## b10 35 -1439.570

treshold\_vif(summary(b10)$r.squared)

## [1] 11.91673

vif(b10)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.855331 1 1.362105  
## log(LotArea) 2.312646 1 1.520739  
## OverallQual 3.977069 1 1.994259  
## OverallCond 1.389785 1 1.178892  
## YearBuilt 3.983699 1 1.995921  
## BsmtFinSF1 2.153663 1 1.467536  
## log(X1stFlrSF) 4.209151 1 2.051622  
## X2ndFlrSF 4.390401 1 2.095328  
## BsmtFullBath 1.857113 1 1.362759  
## FullBath 2.559396 1 1.599811  
## TotRmsAbvGrd 4.010357 1 2.002588  
## KitchenAbvGr 1.507492 1 1.227799  
## Fireplaces 1.537160 1 1.239822  
## GarageCars 3.201199 1 1.789189  
## ScreenPorch 1.074758 1 1.036706  
## PoolArea 1.107622 1 1.052436  
## ExterQual 6.032688 3 1.349227  
## BsmtQual 6.669446 4 1.267685  
## KitchenQual 5.114326 3 1.312597  
## GarageType 4.833162 6 1.140300  
## f.LotArea 1.384153 1 1.176500

b12<- 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+HasPorch\_binary  
 , data = train\_new)  
  
#summary(b11)  
summary(b11)$r.squared

## [1] 0.9164236

anova(b7,b11) # a little better model, but not much, we better keep simpler 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  
## Res.Df RSS Df Sum of Sq F Pr(>F)  
## 1 983 13.465   
## 2 958 13.409 25 0.05591 0.1598 1

AIC(b7,b11)

## df AIC  
## b7 34 -1441.459  
## b11 59 -1395.687

AIC(b7,b71,b3,b31,b2,b21,b1,b11)

## df AIC  
## b7 34 -1441.459  
## b71 98 -1460.390  
## b3 28 -1439.656  
## b31 86 -1436.798  
## b2 25 -1431.238  
## b21 88 -1391.886  
## b1 21 -1386.699  
## b11 59 -1395.687

final\_model<-b3

## Final model metrics

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 0.92 21699.82  
## 2 Model with test\_new sample 0.92 23780.17

## Study the presence of a priori influential data observations

In this case, 26 a priori values where found.

mean\_hat <- mean(hatvalues(final\_model));mean\_hat

## [1] 0.0265748

priori <- which(hatvalues(final\_model)>4\*mean\_hat)  
length(priori)

## [1] 13

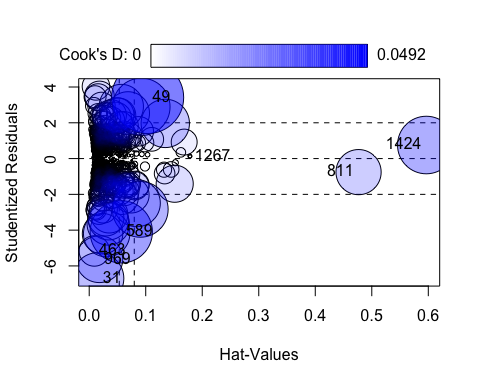
## Study the presence of a posteriori influential values

For studying the posteriori influential values, we have looked to cook’s distance outliers, where 3 influent data are found and are removed from the sample. The best model is then reconstructed on this new data set and compared with its original to see the difference. Comparing the previous coefficients and the current coefficients, we can say that the change has no relevant changes in the coefficients. The R-squared of the model has increased up to 0.94.

betas <- as.data.frame(dfbetas(final\_model))  
betas\_cutoff <- 2 / sqrt(dim(train\_new)[1]);betas\_cutoff

## [1] 0.06274558

influencePlot(final\_model, id=list(n=3, method="noteworthy"))



## StudRes Hat CookD  
## 811 -0.7575219 0.476609386 0.0193620094  
## 31 -6.6975026 0.015620388 0.0252433804  
## 49 3.4127108 0.103405544 0.0492189998  
## 589 -4.0956447 0.056747425 0.0367897392  
## 1267 0.1152849 0.178494741 0.0001070607  
## 969 -5.6247046 0.017934678 0.0207557990  
## 1424 0.7575219 0.596481715 0.0314302390  
## 463 -5.1632300 0.009060561 0.0087996085

llcoo <-c("1424","811","1183")  
# residual outliers  
llres <- which(abs(rstudent(final\_model))>qnorm(0.995));length(llres)

## [1] 26

llrem <- unique(c(rownames(train\_new)[llres],llcoo)); length(llrem)

## [1] 29

llremreg<-which(rownames(train\_new)%in%llrem);llremreg

## [1] 32 55 116 228 241 266 305 331 351 383 558 595 608 621 673  
## [16] 708 723 732 740 753 857 868 914 923 938 950 979 1000

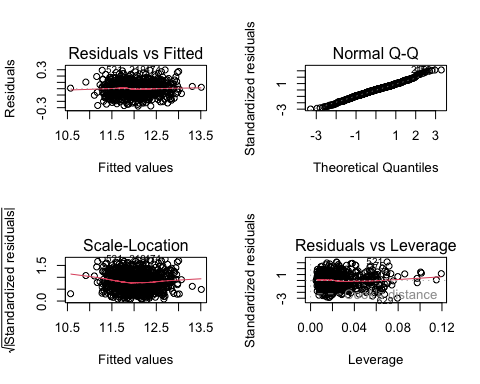
train\_new<-train\_new[-llremreg,]  
  
final\_modelp <- lm(log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + log(X1stFlrSF)+ X2ndFlrSF +  
 BsmtFullBath + FullBath + TotRmsAbvGrd+  
 KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea +ExterQual + BsmtQual + KitchenQual+GarageType  
 , data = train\_new)  
summary(final\_modelp)

##   
## 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 + GarageType, data = train\_new)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.272669 -0.060271 0.002779 0.058393 0.293775   
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.503e+00 4.587e-01 3.276 0.001090 \*\*   
## LotFrontage -6.360e-05 1.734e-04 -0.367 0.713824   
## log(LotArea) 9.752e-02 8.523e-03 11.442 < 2e-16 \*\*\*  
## OverallQual 6.078e-02 4.334e-03 14.024 < 2e-16 \*\*\*  
## OverallCond 5.208e-02 3.190e-03 16.328 < 2e-16 \*\*\*  
## YearBuilt 3.113e-03 2.028e-04 15.344 < 2e-16 \*\*\*  
## BsmtFinSF1 7.583e-05 1.024e-05 7.404 2.90e-13 \*\*\*  
## log(X1stFlrSF) 3.973e-01 2.022e-02 19.647 < 2e-16 \*\*\*  
## X2ndFlrSF 2.434e-04 1.476e-05 16.491 < 2e-16 \*\*\*  
## BsmtFullBath 2.129e-02 7.975e-03 2.670 0.007713 \*\*   
## FullBath 3.005e-02 8.773e-03 3.425 0.000641 \*\*\*  
## TotRmsAbvGrd -1.337e-03 3.814e-03 -0.350 0.726071   
## KitchenAbvGr -7.892e-02 1.785e-02 -4.420 1.10e-05 \*\*\*  
## Fireplaces 2.149e-02 5.777e-03 3.720 0.000211 \*\*\*  
## GarageCars 4.556e-02 7.129e-03 6.390 2.59e-10 \*\*\*  
## ScreenPorch 2.877e-04 6.022e-05 4.777 2.06e-06 \*\*\*  
## PoolArea NA NA NA NA   
## ExterQualFa -6.222e-02 4.413e-02 -1.410 0.158884   
## ExterQualGd -8.247e-03 2.110e-02 -0.391 0.696038   
## ExterQualTA -4.845e-02 2.357e-02 -2.056 0.040063 \*   
## BsmtQualFa -1.706e-01 2.810e-02 -6.070 1.84e-09 \*\*\*  
## BsmtQualGd -9.224e-02 1.567e-02 -5.887 5.45e-09 \*\*\*  
## BsmtQualNo Basement -2.201e-01 2.859e-02 -7.698 3.45e-14 \*\*\*  
## BsmtQualTA -9.929e-02 1.848e-02 -5.372 9.77e-08 \*\*\*  
## KitchenQualFa -1.044e-01 3.004e-02 -3.475 0.000534 \*\*\*  
## KitchenQualGd -6.519e-02 1.575e-02 -4.140 3.79e-05 \*\*\*  
## KitchenQualTA -9.318e-02 1.766e-02 -5.275 1.65e-07 \*\*\*  
## GarageTypeAttchd 8.121e-02 5.028e-02 1.615 0.106607   
## GarageTypeBasment 1.136e-01 5.753e-02 1.975 0.048544 \*   
## GarageTypeBuiltIn 8.568e-02 5.216e-02 1.643 0.100806   
## GarageTypeCarPort 4.783e-03 6.365e-02 0.075 0.940116   
## GarageTypeDetchd 9.476e-02 5.009e-02 1.892 0.058836 .   
## GarageTypeNo Garage 9.644e-02 5.399e-02 1.786 0.074382 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.09572 on 956 degrees of freedom  
## Multiple R-squared: 0.9407, Adjusted R-squared: 0.9387   
## F-statistic: 488.8 on 31 and 956 DF, p-value: < 2.2e-16

final\_modelp <- step( final\_modelp, k=log(nrow(train\_new)))

## Start: AIC=-4448.2  
## log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + ExterQual + BsmtQual + KitchenQual +   
## GarageType  
##   
##   
## Step: AIC=-4448.2  
## log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + ExterQual + BsmtQual + KitchenQual + GarageType  
##   
## Df Sum of Sq RSS AIC  
## - GarageType 6 0.0965 8.8557 -4478.8  
## - ExterQual 3 0.1215 8.8808 -4455.3  
## - TotRmsAbvGrd 1 0.0011 8.7604 -4455.0  
## - LotFrontage 1 0.0012 8.7605 -4455.0  
## <none> 8.7593 -4448.2  
## - BsmtFullBath 1 0.0653 8.8246 -4447.8  
## - FullBath 1 0.1075 8.8667 -4443.1  
## - Fireplaces 1 0.1268 8.8860 -4440.9  
## - KitchenQual 3 0.2594 9.0187 -4440.1  
## - KitchenAbvGr 1 0.1790 8.9383 -4435.1  
## - ScreenPorch 1 0.2091 8.9683 -4431.8  
## - GarageCars 1 0.3741 9.1334 -4413.8  
## - BsmtQual 4 0.6597 9.4189 -4404.0  
## - BsmtFinSF1 1 0.5022 9.2615 -4400.0  
## - log(LotArea) 1 1.1996 9.9589 -4328.3  
## - OverallQual 1 1.8019 10.5612 -4270.3  
## - YearBuilt 1 2.1573 10.9166 -4237.6  
## - OverallCond 1 2.4428 11.2021 -4212.1  
## - X2ndFlrSF 1 2.4917 11.2509 -4207.8  
## - log(X1stFlrSF) 1 3.5367 12.2960 -4120.0  
##   
## Step: AIC=-4478.76  
## log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + ExterQual + BsmtQual + KitchenQual  
##   
## Df Sum of Sq RSS AIC  
## - ExterQual 3 0.1200 8.9757 -4486.1  
## - TotRmsAbvGrd 1 0.0003 8.8560 -4485.6  
## - LotFrontage 1 0.0020 8.8577 -4485.4  
## <none> 8.8557 -4478.8  
## - BsmtFullBath 1 0.0666 8.9223 -4478.3  
## - FullBath 1 0.1050 8.9607 -4474.0  
## - Fireplaces 1 0.1285 8.9843 -4471.4  
## - KitchenQual 3 0.2550 9.1107 -4471.4  
## - ScreenPorch 1 0.2088 9.0645 -4462.6  
## - KitchenAbvGr 1 0.2288 9.0845 -4460.5  
## - BsmtQual 4 0.6756 9.5313 -4433.7  
## - BsmtFinSF1 1 0.4904 9.3461 -4432.4  
## - GarageCars 1 0.5052 9.3609 -4430.8  
## - log(LotArea) 1 1.1605 10.0162 -4364.0  
## - OverallQual 1 1.8836 10.7393 -4295.1  
## - YearBuilt 1 2.4228 11.2785 -4246.7  
## - OverallCond 1 2.4734 11.3291 -4242.3  
## - X2ndFlrSF 1 2.5827 11.4384 -4232.8  
## - log(X1stFlrSF) 1 3.5721 12.4278 -4150.8  
##   
## Step: AIC=-4486.15  
## log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + BsmtQual + KitchenQual  
##   
## Df Sum of Sq RSS AIC  
## - TotRmsAbvGrd 1 0.0008 8.9764 -4493.0  
## - LotFrontage 1 0.0026 8.9783 -4492.8  
## <none> 8.9757 -4486.1  
## - BsmtFullBath 1 0.0732 9.0489 -4485.0  
## - FullBath 1 0.1035 9.0791 -4481.7  
## - Fireplaces 1 0.1145 9.0902 -4480.5  
## - ScreenPorch 1 0.2059 9.1816 -4470.6  
## - KitchenAbvGr 1 0.2214 9.1971 -4469.0  
## - KitchenQual 3 0.4653 9.4410 -4456.9  
## - BsmtFinSF1 1 0.4570 9.4327 -4444.0  
## - GarageCars 1 0.5065 9.4822 -4438.8  
## - BsmtQual 4 0.7215 9.6972 -4437.3  
## - log(LotArea) 1 1.1473 10.1230 -4374.2  
## - OverallQual 1 2.4280 11.4037 -4256.5  
## - OverallCond 1 2.5353 11.5110 -4247.2  
## - X2ndFlrSF 1 2.6230 11.5987 -4239.7  
## - YearBuilt 1 2.6529 11.6286 -4237.2  
## - log(X1stFlrSF) 1 3.6618 12.6375 -4155.0  
##   
## Step: AIC=-4492.96  
## log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + KitchenAbvGr + Fireplaces + GarageCars + ScreenPorch +   
## BsmtQual + KitchenQual  
##   
## Df Sum of Sq RSS AIC  
## - LotFrontage 1 0.0028 8.9793 -4499.5  
## <none> 8.9764 -4493.0  
## - BsmtFullBath 1 0.0729 9.0494 -4491.9  
## - FullBath 1 0.1028 9.0792 -4488.6  
## - Fireplaces 1 0.1144 9.0908 -4487.3  
## - ScreenPorch 1 0.2074 9.1839 -4477.3  
## - KitchenAbvGr 1 0.2401 9.2165 -4473.8  
## - KitchenQual 3 0.4651 9.4416 -4463.7  
## - BsmtFinSF1 1 0.4734 9.4499 -4449.1  
## - GarageCars 1 0.5060 9.4824 -4445.7  
## - BsmtQual 4 0.7264 9.7028 -4443.7  
## - log(LotArea) 1 1.1470 10.1235 -4381.0  
## - OverallQual 1 2.4318 11.4082 -4263.0  
## - OverallCond 1 2.5355 11.5120 -4254.1  
## - YearBuilt 1 2.6529 11.6294 -4244.0  
## - log(X1stFlrSF) 1 4.7935 13.7699 -4077.1  
## - X2ndFlrSF 1 4.7992 13.7756 -4076.7  
##   
## Step: AIC=-4499.54  
## log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond + YearBuilt +   
## BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + KitchenAbvGr + Fireplaces + GarageCars + ScreenPorch +   
## BsmtQual + KitchenQual  
##   
## Df Sum of Sq RSS AIC  
## <none> 8.9793 -4499.5  
## - BsmtFullBath 1 0.0736 9.0529 -4498.4  
## - FullBath 1 0.1030 9.0823 -4495.2  
## - Fireplaces 1 0.1162 9.0955 -4493.7  
## - ScreenPorch 1 0.2060 9.1853 -4484.0  
## - KitchenAbvGr 1 0.2392 9.2185 -4480.5  
## - KitchenQual 3 0.4681 9.4474 -4470.0  
## - BsmtFinSF1 1 0.4753 9.4546 -4455.5  
## - GarageCars 1 0.5034 9.4826 -4452.6  
## - BsmtQual 4 0.7242 9.7035 -4450.5  
## - log(LotArea) 1 1.4681 10.4474 -4356.8  
## - OverallQual 1 2.4310 11.4103 -4269.7  
## - OverallCond 1 2.5331 11.5124 -4260.9  
## - YearBuilt 1 2.6583 11.6376 -4250.2  
## - X2ndFlrSF 1 4.8079 13.7872 -4082.8  
## - log(X1stFlrSF) 1 4.8760 13.8552 -4077.9

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



coef\_orig <- coef(final\_model)  
coef\_filt <- coef(final\_modelp)  
  
#common variables  
var\_com <- intersect(names(coef\_orig), names(coef\_filt))  
  
coef\_orig <- coef\_orig[var\_com]  
coef\_filt <- coef\_filt[var\_com]  
  
coef\_comparison <- data.frame(Variable = names(coef\_orig),  
 Original\_coefficient = coef\_orig,  
 Filtered\_coefficient = coef\_filt)  
print(coef\_comparison)

## Variable Original\_coefficient  
## (Intercept) (Intercept) 1.820753e+00  
## log(LotArea) log(LotArea) 8.956534e-02  
## OverallQual OverallQual 6.664449e-02  
## OverallCond OverallCond 5.254212e-02  
## YearBuilt YearBuilt 3.030376e-03  
## BsmtFinSF1 BsmtFinSF1 7.839739e-05  
## log(X1stFlrSF) log(X1stFlrSF) 3.879723e-01  
## X2ndFlrSF X2ndFlrSF 2.327846e-04  
## BsmtFullBath BsmtFullBath 2.344988e-02  
## FullBath FullBath 2.872197e-02  
## KitchenAbvGr KitchenAbvGr -6.904739e-02  
## Fireplaces Fireplaces 2.475002e-02  
## GarageCars GarageCars 4.841963e-02  
## ScreenPorch ScreenPorch 2.960222e-04  
## BsmtQualFa BsmtQualFa -1.541059e-01  
## BsmtQualGd BsmtQualGd -9.552989e-02  
## BsmtQualNo Basement BsmtQualNo Basement -2.183787e-01  
## BsmtQualTA BsmtQualTA -1.057755e-01  
## KitchenQualFa KitchenQualFa -6.398483e-02  
## KitchenQualGd KitchenQualGd -4.245788e-02  
## KitchenQualTA KitchenQualTA -7.451015e-02  
## Filtered\_coefficient  
## (Intercept) 1.589139e+00  
## log(LotArea) 9.219227e-02  
## OverallQual 6.658310e-02  
## OverallCond 5.175633e-02  
## YearBuilt 3.126180e-03  
## BsmtFinSF1 7.241335e-05  
## log(X1stFlrSF) 3.934097e-01  
## X2ndFlrSF 2.401428e-04  
## BsmtFullBath 2.250707e-02  
## FullBath 2.895870e-02  
## KitchenAbvGr -8.620365e-02  
## Fireplaces 2.027016e-02  
## GarageCars 4.260217e-02  
## ScreenPorch 2.840820e-04  
## BsmtQualFa -1.771345e-01  
## BsmtQualGd -9.503964e-02  
## BsmtQualNo Basement -2.248469e-01  
## BsmtQualTA -1.070349e-01  
## KitchenQualFa -1.134850e-01  
## KitchenQualGd -6.608734e-02  
## KitchenQualTA -1.099259e-01

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

## Model RSQRT RMSE  
## 1 Model with train\_new sample 0.94 19429.18  
## 2 Model with test\_new sample 0.92 23865.33

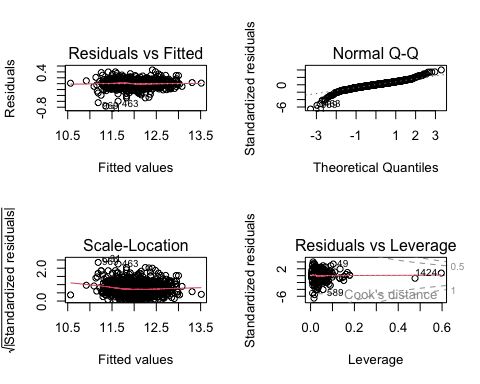
## Interpret the Model

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   
## -0.76387 -0.05992 0.00430 0.06456 0.46744   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.821e+00 5.067e-01 3.593 0.000343 \*\*\*  
## LotFrontage -6.427e-05 1.983e-04 -0.324 0.745964   
## log(LotArea) 8.957e-02 1.010e-02 8.871 < 2e-16 \*\*\*  
## OverallQual 6.664e-02 5.207e-03 12.798 < 2e-16 \*\*\*  
## OverallCond 5.254e-02 3.835e-03 13.701 < 2e-16 \*\*\*  
## YearBuilt 3.030e-03 2.245e-04 13.501 < 2e-16 \*\*\*  
## BsmtFinSF1 7.840e-05 1.245e-05 6.299 4.51e-10 \*\*\*  
## log(X1stFlrSF) 3.880e-01 2.410e-02 16.100 < 2e-16 \*\*\*  
## X2ndFlrSF 2.328e-04 1.743e-05 13.355 < 2e-16 \*\*\*  
## BsmtFullBath 2.345e-02 9.691e-03 2.420 0.015711 \*   
## FullBath 2.872e-02 1.060e-02 2.709 0.006859 \*\*   
## TotRmsAbvGrd -4.670e-04 4.597e-03 -0.102 0.919103   
## KitchenAbvGr -6.905e-02 2.017e-02 -3.423 0.000645 \*\*\*  
## Fireplaces 2.475e-02 6.941e-03 3.566 0.000380 \*\*\*  
## GarageCars 4.842e-02 6.932e-03 6.985 5.22e-12 \*\*\*  
## ScreenPorch 2.960e-04 7.358e-05 4.023 6.18e-05 \*\*\*  
## PoolArea 1.486e-04 1.252e-04 1.187 0.235316   
## ExterQualFa -6.852e-02 5.292e-02 -1.295 0.195700   
## ExterQualGd -2.368e-02 2.547e-02 -0.930 0.352707   
## ExterQualTA -5.527e-02 2.832e-02 -1.952 0.051263 .   
## BsmtQualFa -1.541e-01 3.376e-02 -4.565 5.63e-06 \*\*\*  
## BsmtQualGd -9.553e-02 1.902e-02 -5.023 6.04e-07 \*\*\*  
## BsmtQualNo Basement -2.184e-01 3.461e-02 -6.310 4.20e-10 \*\*\*  
## BsmtQualTA -1.058e-01 2.222e-02 -4.761 2.21e-06 \*\*\*  
## KitchenQualFa -6.398e-02 3.523e-02 -1.816 0.069639 .   
## KitchenQualGd -4.246e-02 1.879e-02 -2.260 0.024049 \*   
## KitchenQualTA -7.451e-02 2.105e-02 -3.540 0.000419 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1175 on 989 degrees of freedom  
## Multiple R-squared: 0.9149, Adjusted R-squared: 0.9127   
## F-statistic: 409.1 on 26 and 989 DF, p-value: < 2.2e-16

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



coefficients <- coef(final\_model)  
coefficients

## (Intercept) LotFrontage log(LotArea) OverallQual   
## 1.820753e+00 -6.427222e-05 8.956534e-02 6.664449e-02   
## OverallCond YearBuilt BsmtFinSF1 log(X1stFlrSF)   
## 5.254212e-02 3.030376e-03 7.839739e-05 3.879723e-01   
## X2ndFlrSF BsmtFullBath FullBath TotRmsAbvGrd   
## 2.327846e-04 2.344988e-02 2.872197e-02 -4.670149e-04   
## KitchenAbvGr Fireplaces GarageCars ScreenPorch   
## -6.904739e-02 2.475002e-02 4.841963e-02 2.960222e-04   
## PoolArea ExterQualFa ExterQualGd ExterQualTA   
## 1.486301e-04 -6.852162e-02 -2.368096e-02 -5.526604e-02   
## BsmtQualFa BsmtQualGd BsmtQualNo Basement BsmtQualTA   
## -1.541059e-01 -9.552989e-02 -2.183787e-01 -1.057755e-01   
## KitchenQualFa KitchenQualGd KitchenQualTA   
## -6.398483e-02 -4.245788e-02 -7.451015e-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).

## Annex