Assignment 1

Alicia Chimeno & Silvia Ferrer

2023-11-19

# Clear space and Libraries

# Load dataset

# Explore and Preproces the dataset

We start by looking at an overview of our dataset.

dim(train)

## [1] 1460 81

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 ...

## 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

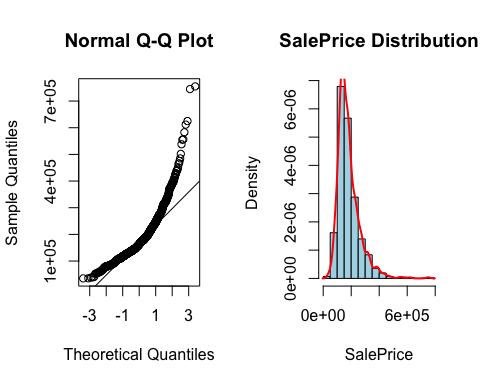
shapiro.test(train$SalePrice)

##   
## 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)



## 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])]

# 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\_description <- function(variable, n\_breaks) {  
 #cat("Summary:\n")  
 #print(summary(variable))  
 cat("Count of missing values:",sum(is.na(variable)),"\n")  
 column\_name <- sub(".+\\$", "", deparse(substitute(variable)))  
 hist(variable, breaks = n\_breaks, freq = F, main = paste("Histogram of", column\_name), xlab = column\_name)  
 curve(dnorm(x, mean(variable), sd(variable)), add = T)  
 print(shapiro.test(variable))  
}  
  
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")  
 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")  
 abline(h = severe\_ub, col = "red", lwd = 2.5)  
 abline(h = severe\_lb, col = "red", lwd = 2.5)  
}  
  
train2$univ\_outl\_count <- 0 # new column count outliers - initialize  
  
update\_outliers\_count <- function(dataframe, column\_name) {  
 df <- dataframe  
 sm <- summary(df[[column\_name]])  
 iqr <- sm["3rd Qu."] - sm["1st Qu."]  
 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. It 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  
summary(train2)  
#names(train2)

## 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 (number of garages, rooms…)

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

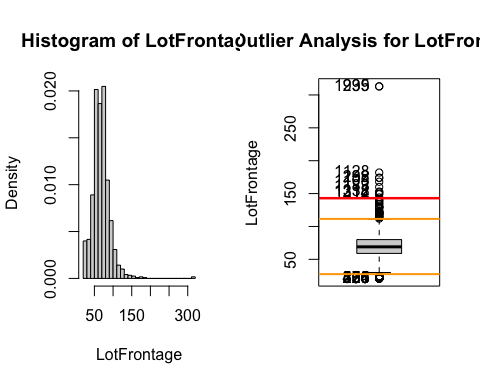
par(mfrow=c(1,2))  
numeric\_description(train2$LotFrontage, 30) # not normal distribution

## 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(1,3))  
numeric\_description(train2$LotArea, 20) # not normal distribution

## Count of missing values: 0

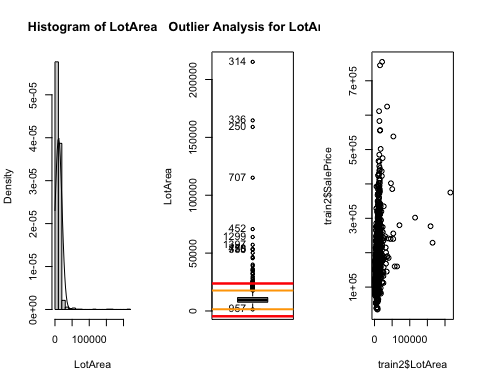
##   
## Shapiro-Wilk normality test  
##   
## data: variable  
## W = 0.35106, p-value < 2.2e-16

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)

### “OverallQual” (cont)

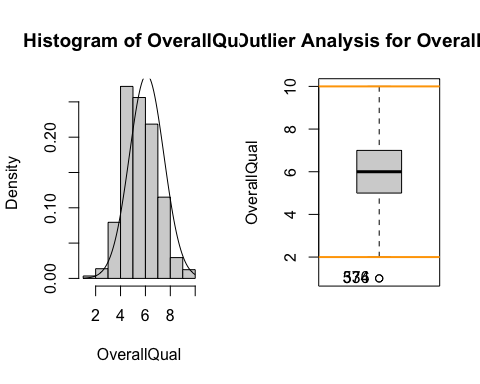
par(mfrow=c(1,2))  
numeric\_description(train2$OverallQual, 10) # not normal distribution

## 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)

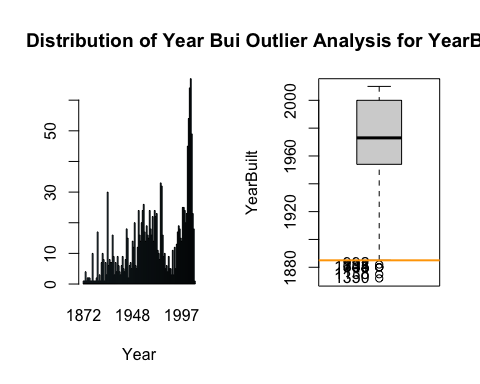
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1872 1954 1973 1971 2000 2010

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

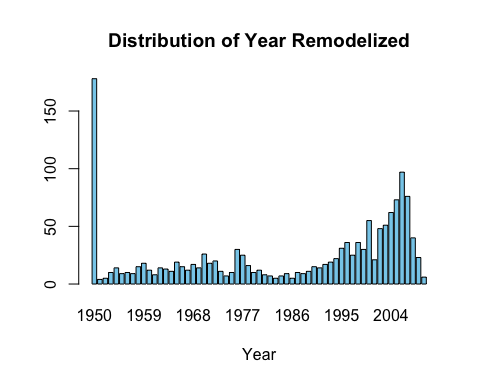
train2 <- update\_outliers\_count(train2, "YearBuilt")

### “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")

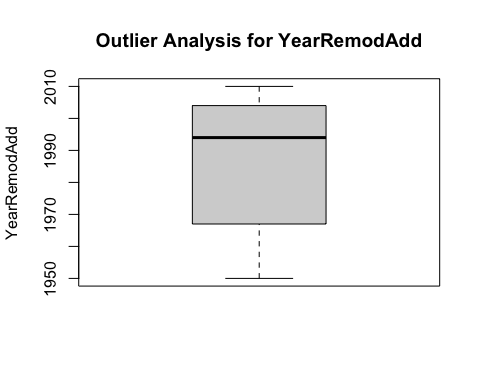


sum(is.na(train2$YearRemodAdd))

## [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)

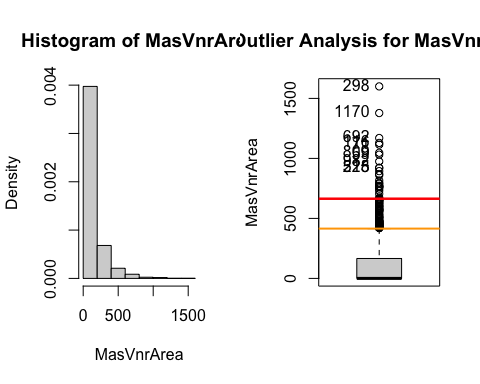
par(mfrow=c(1,2))  
numeric\_description(train2$MasVnrArea, 10)# not normal distribution

## 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)

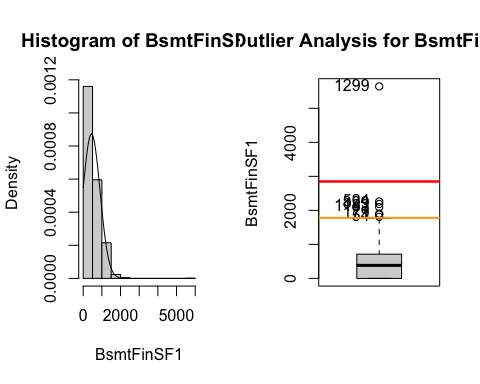
par(mfrow=c(1,2))  
numeric\_description(train2$BsmtFinSF1, 10)# not normal distribution

## 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)

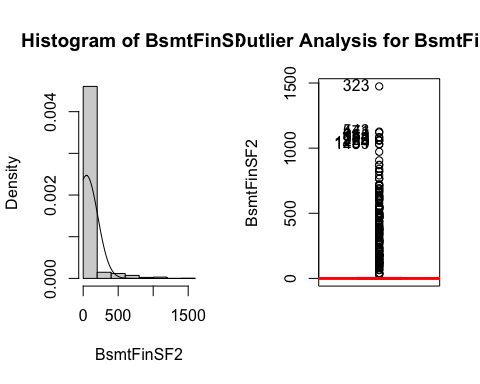
par(mfrow=c(1,2))  
numeric\_description(train2$BsmtFinSF2, 10) # not normal distribution

## 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)

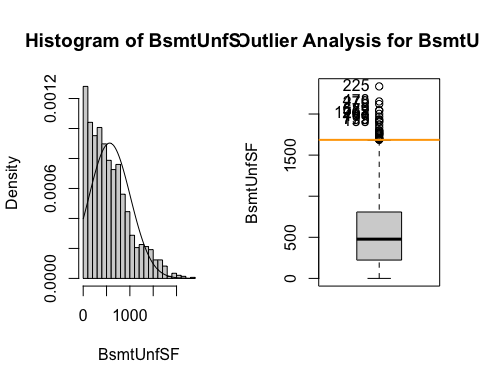
par(mfrow=c(1,2))  
numeric\_description(train2$BsmtUnfSF, 20) # not normal distribution

## 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)

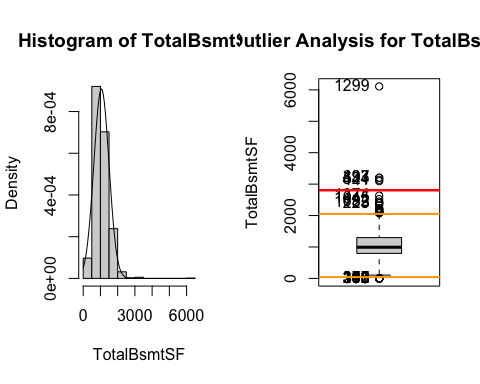
par(mfrow=c(1,2))  
numeric\_description(train2$TotalBsmtSF, 10)# not normal distribution

## 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)

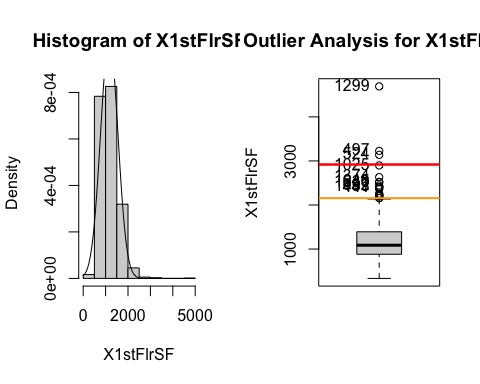
par(mfrow=c(1,2))  
numeric\_description(train2$X1stFlrSF, 10) # not normal distribution

## 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)

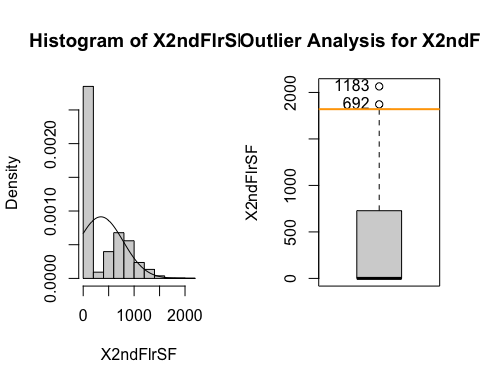
par(mfrow=c(1,2))  
numeric\_description(train2$X2ndFlrSF, 10)# not normal distribution

## 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)

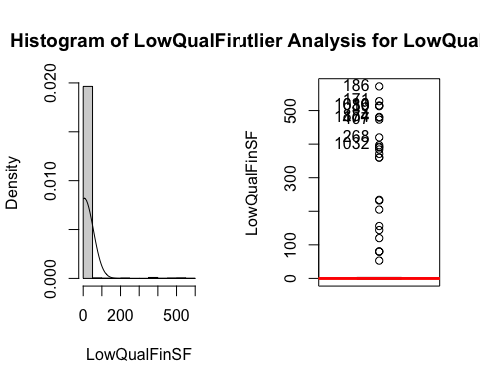
par(mfrow=c(1,2))  
numeric\_description(train2$LowQualFinSF, 10) # not normal distribution

## 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   
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)

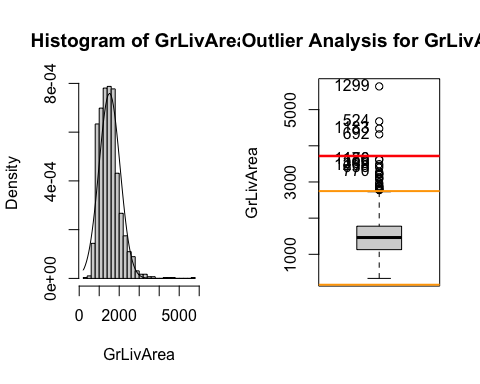
par(mfrow=c(1,2))  
numeric\_description(train2$GrLivArea, 20) # not normal distribution

## 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")

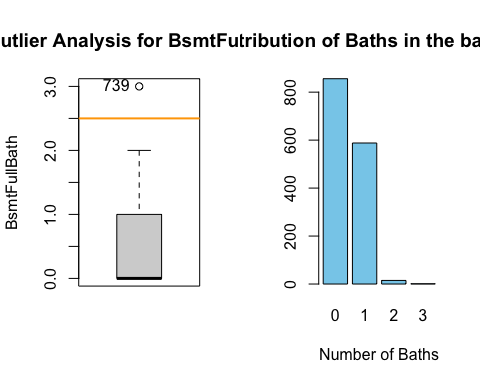
## Number of mild outliers: 1

## 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")



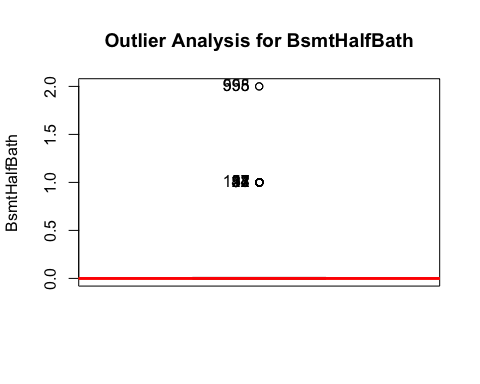
### BsmtHalfBath - # half baths in the basement (int)

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

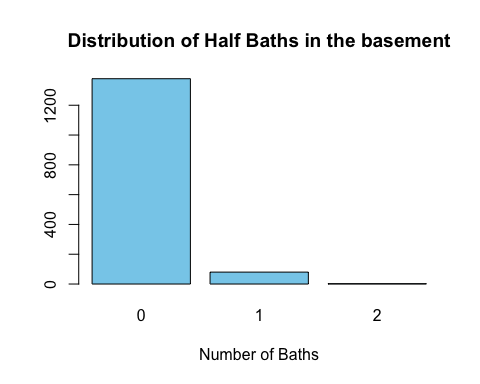
nearZeroVar(train2$BsmtHalfBath, saveMetrics = TRUE) # Variable not with near zero variance

## 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

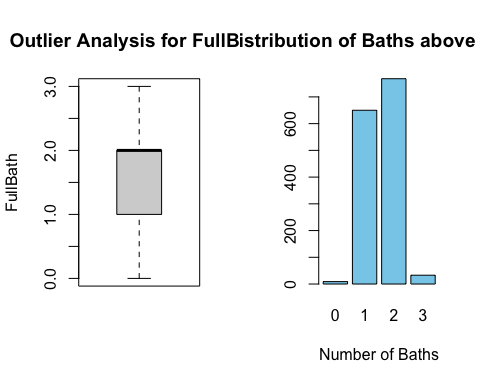
train2 <- update\_outliers\_count(train2, "FullBath")  
table(train2$FullBath)

##   
## 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

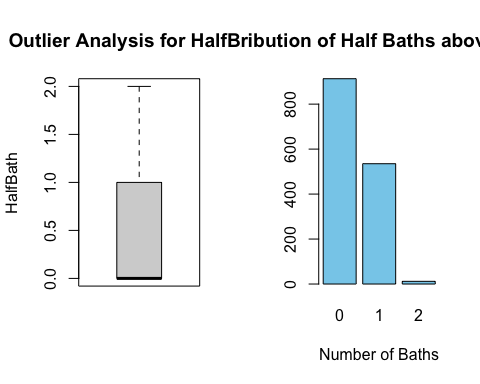
train2 <- update\_outliers\_count(train2, "HalfBath")  
table(train2$HalfBath)

##   
## 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

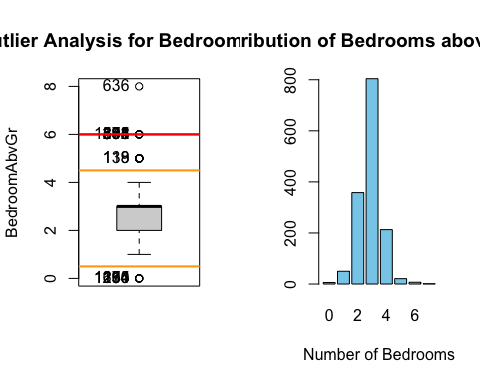
train2 <- update\_outliers\_count(train2, "BedroomAbvGr")  
table(train2$BedroomAbvGr)

##   
## 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")

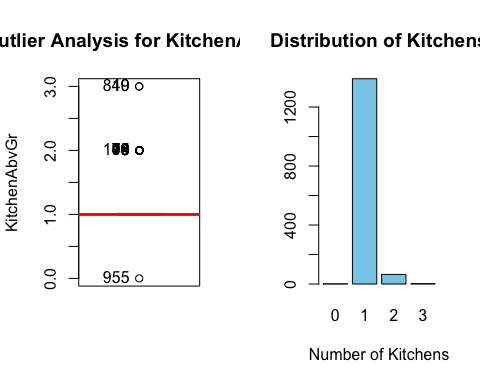
## Number of mild outliers: 68

## 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

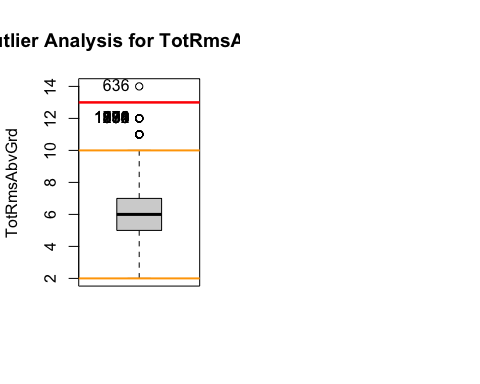
analyze\_outliers(train2,"TotRmsAbvGrd")

## 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))

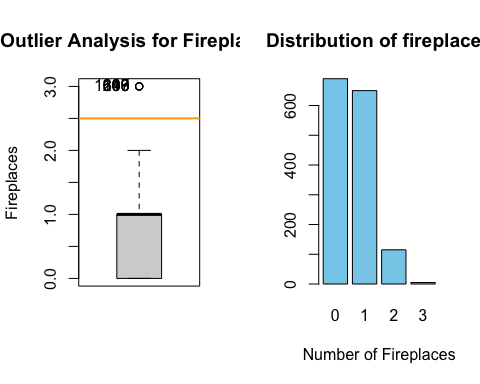
## [1] 0

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

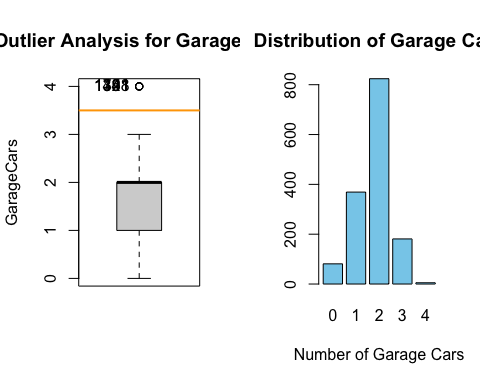
## [1] 0

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)

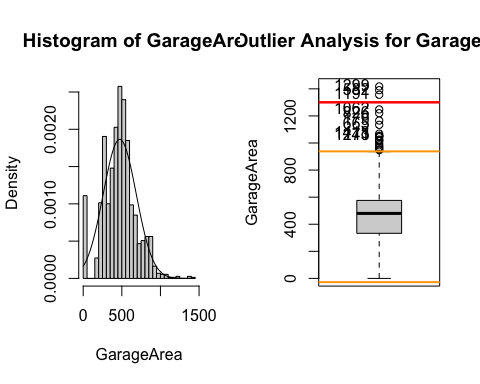
par(mfrow=c(1,2))  
numeric\_description(train2$GarageArea, 30) # not normal distribution

## 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

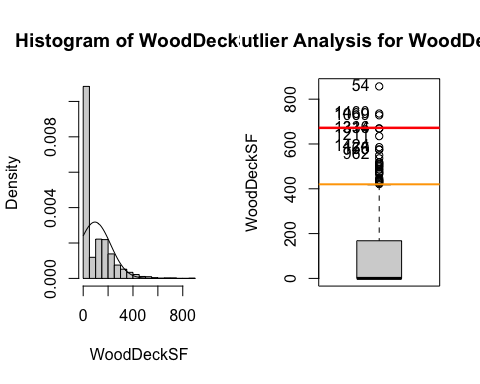
par(mfrow=c(1,2))  
numeric\_description(train2$WoodDeckSF, 30) # not normal distribution

## 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



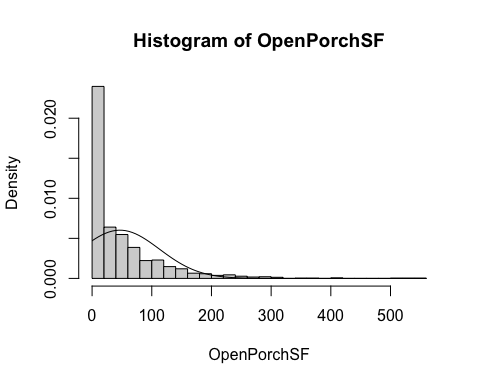
## Number of severe outliers: 3

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

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

numeric\_description(train2$OpenPorchSF, 30)

## Count of missing values: 0



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

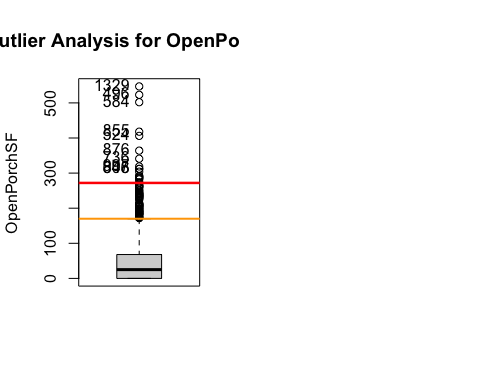
par(mfrow=c(1,2))  
analyze\_outliers(train2,"OpenPorchSF")

## 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)

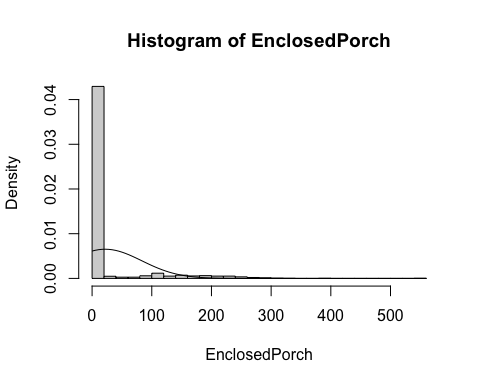
##   
## 0 1   
## 656 804



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

numeric\_description(train2$EnclosedPorch, 30)

## Count of missing values: 0



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

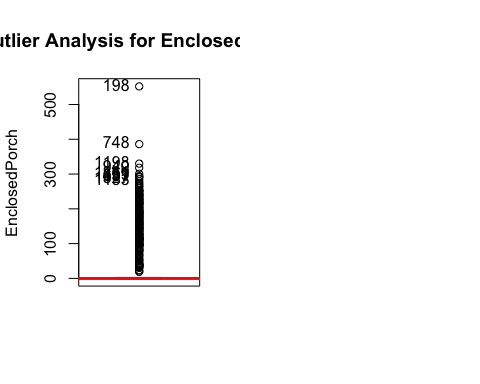
par(mfrow=c(1,2))  
analyze\_outliers(train2,"EnclosedPorch")

## 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)

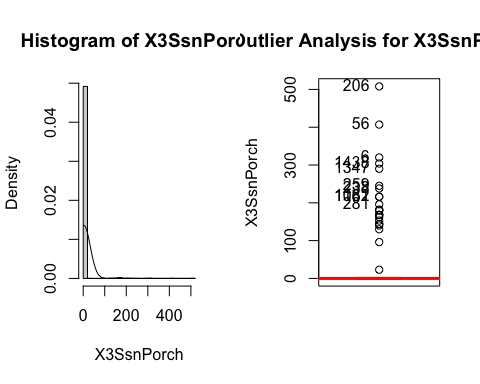
par(mfrow=c(1,2))  
numeric\_description(train2$X3SsnPorch, 30)

## 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)

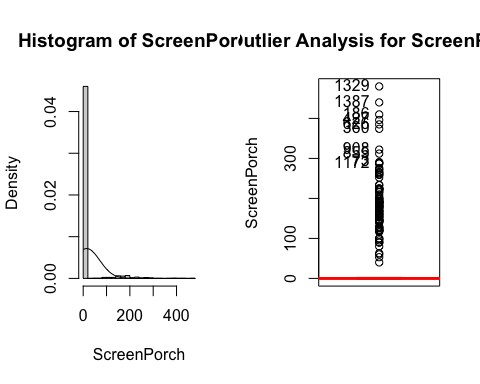
par(mfrow=c(1,2))  
numeric\_description(train2$ScreenPorch, 30) # not normally distributed

## 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)

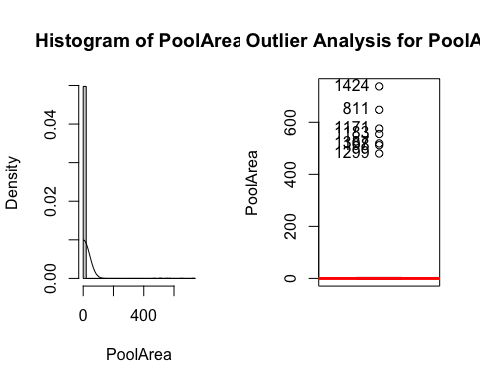
par(mfrow=c(1,2))  
numeric\_description(train2$PoolArea, 30)# not normally distributed

## 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)

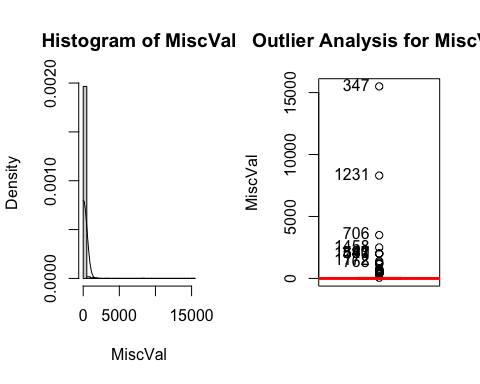
par(mfrow=c(1,2))  
numeric\_description(train2$MiscVal, 30) # not normally distributed

## 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



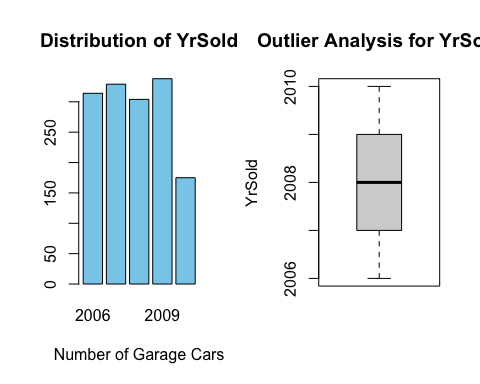
## Number of severe outliers: 52

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

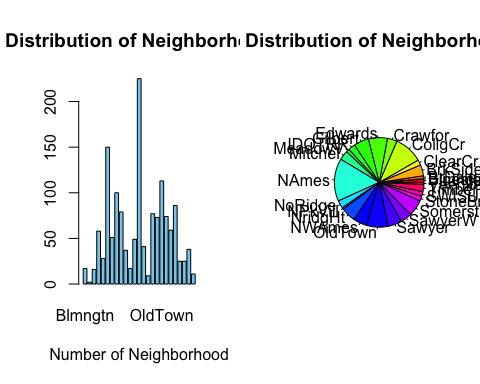
Categorical variables are analyzed.

### Neighborhood

par(mfrow=c(1,2))  
sum(is.na(train2$Neighborhood)) #0

## [1] 0

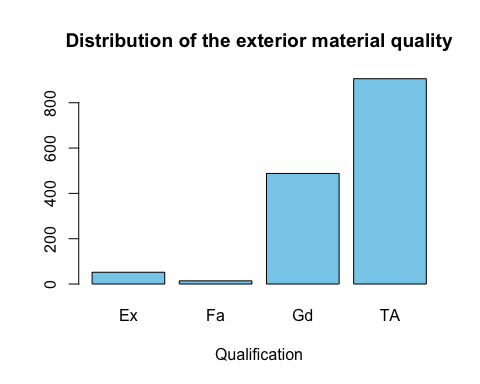
#table(train2$Neighborhood)  
barplot(table(train2$Neighborhood), main = "Distribution of Neighborhood",xlab = "Number of Neighborhood",col = "skyblue")  
pie(table(train2$Neighborhood), main = "Distribution of Neighborhood", col = rainbow(length(levels(train2$Neighborhood))))

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

par(mfrow=c(1,1))  
sum(is.na(train2$ExterQual)) #0

## [1] 0

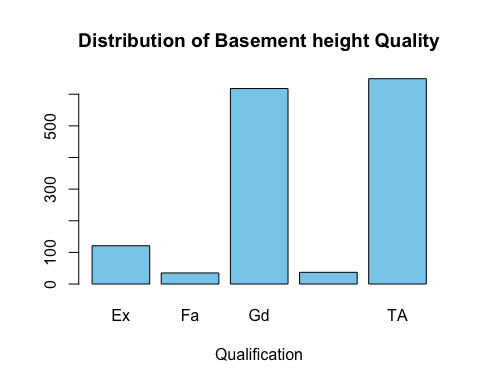
#table(train2$ExterQual)  
barplot(table(train2$ExterQual), main = "Distribution of the exterior material quality",xlab = "Qualification",col = "skyblue")

 ### BsmtQual - Evaluates the height of the basement

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

## [1] 0

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

 ### KitchenQual - Kitchen quality

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

## [1] 0

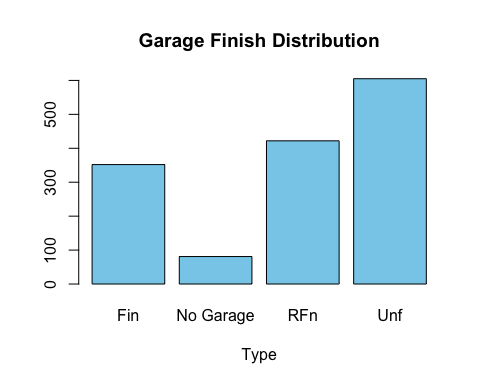
#table(train2$KitchenQual)  
barplot(table(train2$KitchenQual), main = "Distribution of Kitchen Quality",xlab = "Qualification",col = "skyblue")

 ### GarageFinish - Interior finish of the garage

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

## [1] 0

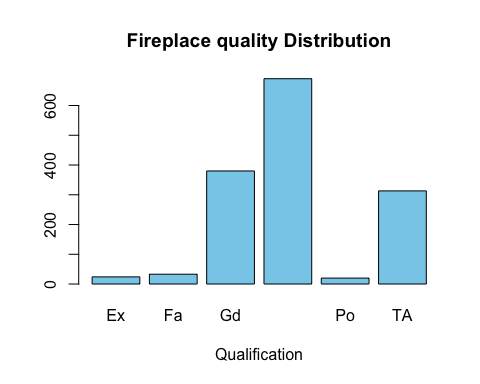
#table(train2$GarageFinish)  
barplot(table(train2$GarageFinish), main = "Garage Finish Distribution",xlab = "Type",col = "skyblue")

 ### FireplaceQu - Fireplace quality

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

## [1] 0

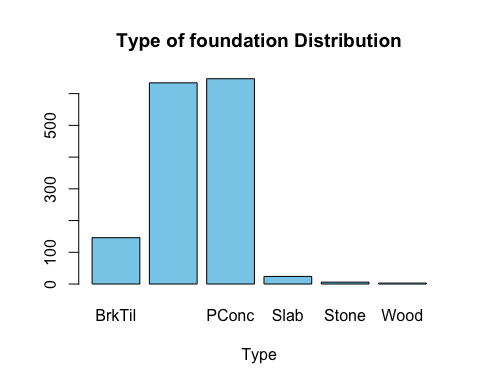
#table(train2$FireplaceQu)  
barplot(table(train2$FireplaceQu), main = "Fireplace quality Distribution",xlab = "Qualification",col = "skyblue")

 ### Foundation - Type of foundation

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

## [1] 0

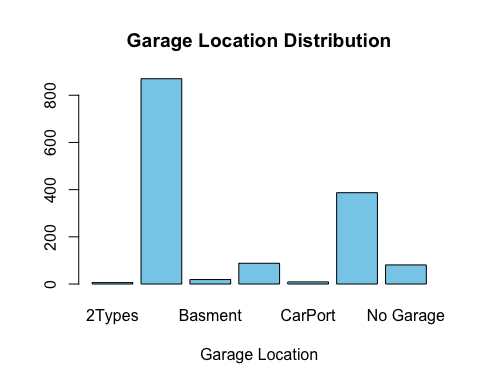
#table(train2$Foundation)  
barplot(table(train2$Foundation), main = "Type of foundation Distribution",xlab = "Type",col = "skyblue")

 ### GarageType - Garage location

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

## [1] 0

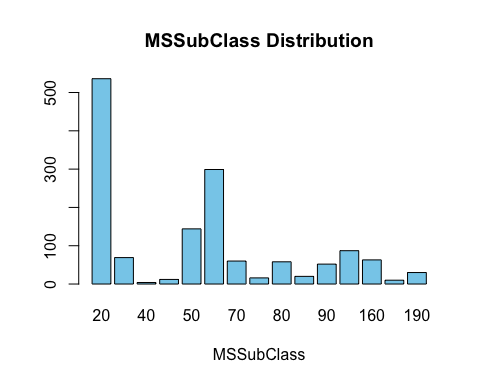
#table(train2$GarageType)  
barplot(table(train2$GarageType), main = "Garage Location Distribution",xlab = "Garage Location",col = "skyblue")

 ### MSSubClass

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

## [1] 0

#table(train2$MSSubClass) # +classes  
barplot(table(train2$MSSubClass), main = "MSSubClass Distribution",xlab = "MSSubClass",col = "skyblue")

 ## 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.

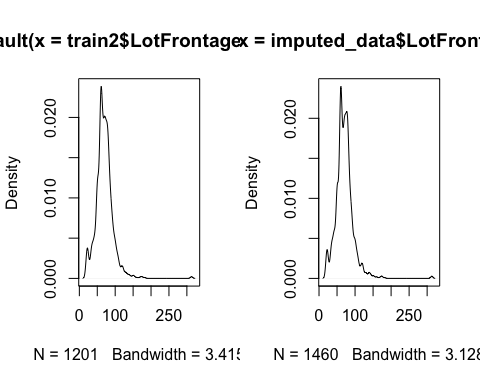
# LotFrontage validation  
summary(imputed\_data$LotFrontage)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 21.00 60.00 70.00 70.67 80.00 313.00

summary(train2$LotFrontage)

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

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.

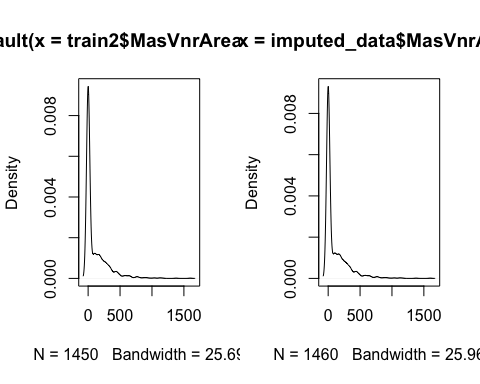
# MasVnrArea validation  
summary(imputed\_data$MasVnrArea)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0 0.0 0.0 103.8 166.0 1600.0

summary(train2$MasVnrArea)

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

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)

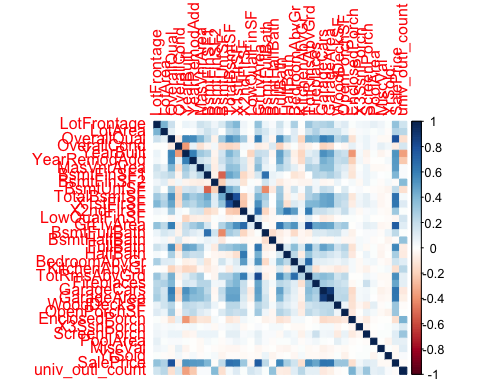
## [1] 10

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, ]  
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

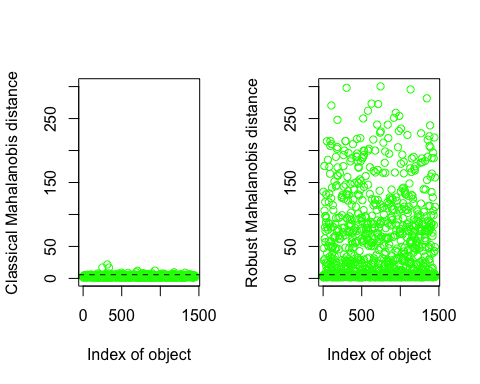
## Multivariate outliers

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")



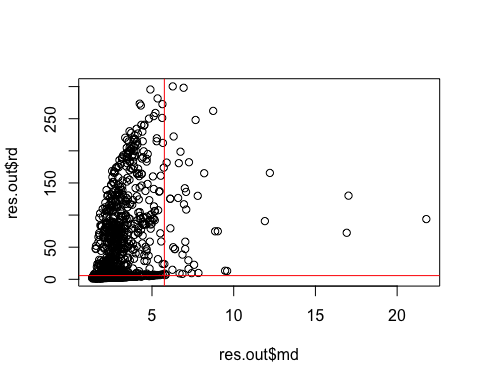
which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff))

## 54 103 179 186 250 305 314 333 336 385 441 452 496 497 582 584   
## 54 103 179 186 249 304 313 332 335 384 440 451 495 496 579 581   
## 665 667 692 707 748 770 779 804 808 855 869 895 899 935 1012 1025   
## 661 663 688 703 744 766 775 800 804 850 864 890 894 930 1007 1020   
## 1046 1047 1049 1062 1170 1174 1183 1185 1191 1269 1293 1329 1384 1397   
## 1041 1042 1044 1057 1165 1169 1178 1180 1186 1263 1287 1322 1377 1389

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

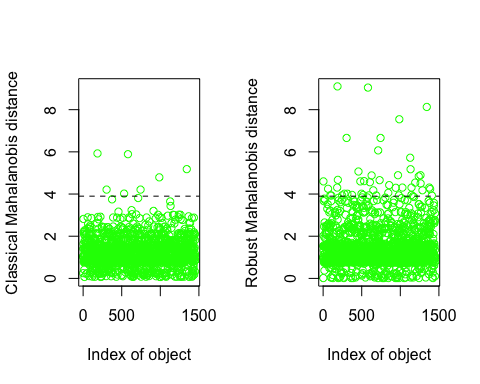
## [1] 46

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")

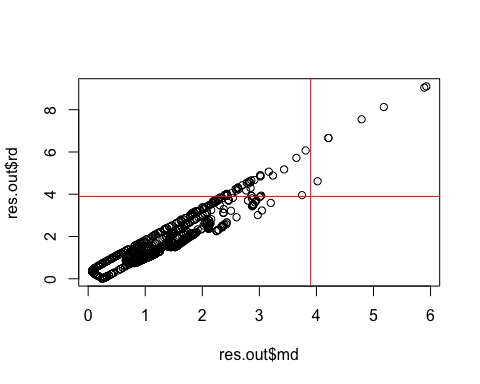


# take off m outliers  
#imputed\_data <- imputed\_data[-which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff)),]

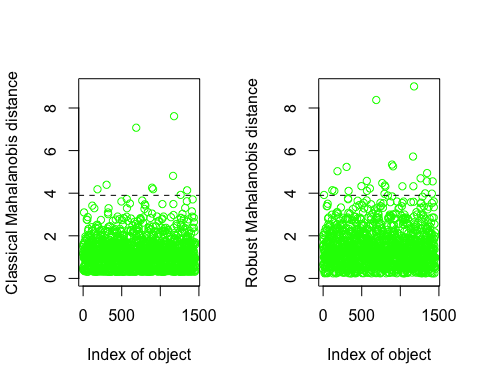
# 2. Generally the houses built latetly are the ones with most quality  
res.out <- Moutlier(imputed\_data[,c("YearBuilt","OverallQual")], quantile = 0.9995, col="green")



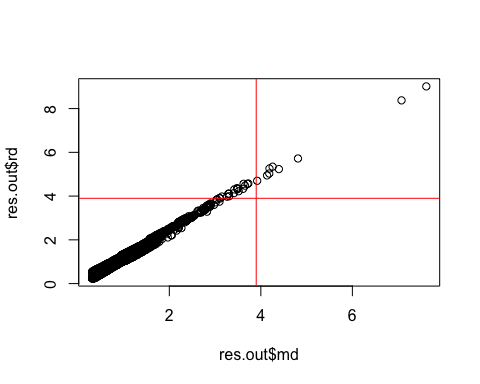
which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff))  
par(mfrow=c(1,1))  
plot( res.out$md, res.out$rd )  
abline(h=res.out$cutoff, col="red")  
abline(v=res.out$cutoff, col="red")



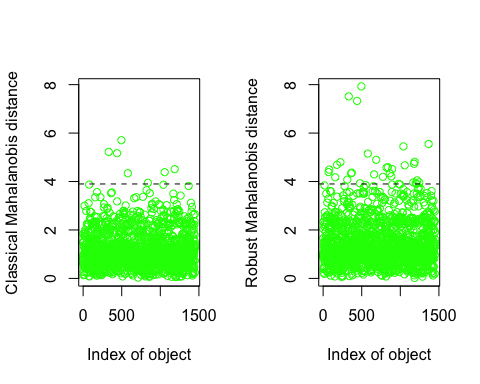
# 3. Generally the houses with more area are the ones with more rooms  
res.out <- Moutlier(imputed\_data[,c("GrLivArea","TotRmsAbvGrd")], quantile = 0.9995, col="green")



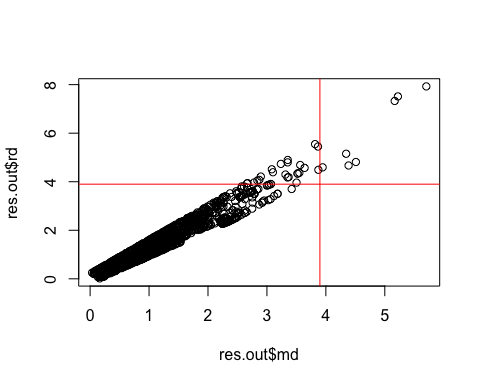
which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff))  
par(mfrow=c(1,1))  
plot( res.out$md, res.out$rd )  
abline(h=res.out$cutoff, col="red")  
abline(v=res.out$cutoff, col="red")



# 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")

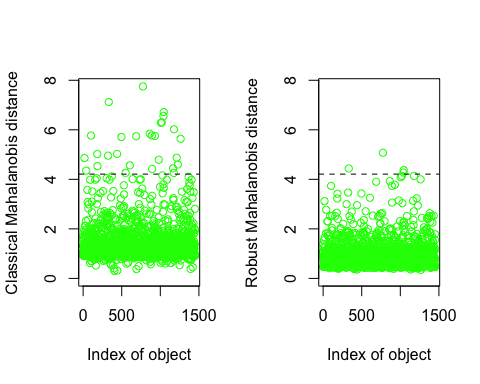


which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff))  
par(mfrow=c(1,1))  
plot( res.out$md, res.out$rd )  
abline(h=res.out$cutoff, col="red")  
abline(v=res.out$cutoff, col="red")

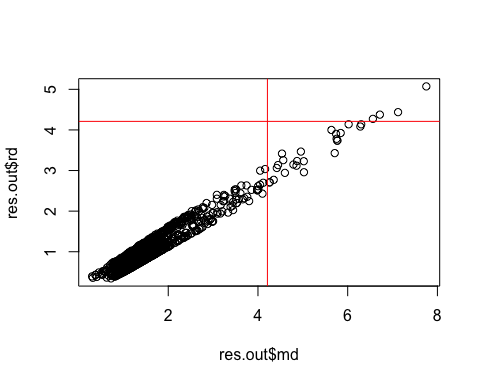


# 5. Moutliers about total areas zones  
res.out <- Moutlier(imputed\_data[,c("X1stFlrSF","TotalBsmtSF","GrLivArea")], quantile = 0.9995, col="green") #GarageArea, si se lo sumo hay menos outliers

## Warning in covMcd(X): The covariance matrix has become singular during  
## the iterations of the MCD algorithm.  
## There are 657 observations (in the entire dataset of 1452 obs.) lying  
## on the plane with equation 0.70711 (x\_i1-m\_1) + -1.4268e-15 (x\_i2-m\_2)  
## + -0.70711 (x\_i3-m\_3) = 0 with (m\_1,m\_2) the mean of these  
## observations.



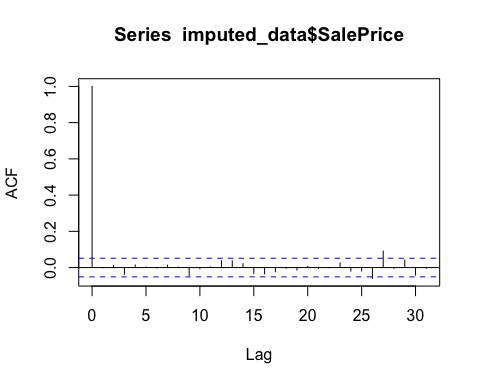
which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff))  
par(mfrow=c(1,1))  
plot( res.out$md, res.out$rd )  
abline(h=res.out$cutoff, col="red")  
abline(v=res.out$cutoff, col="red")



# 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)



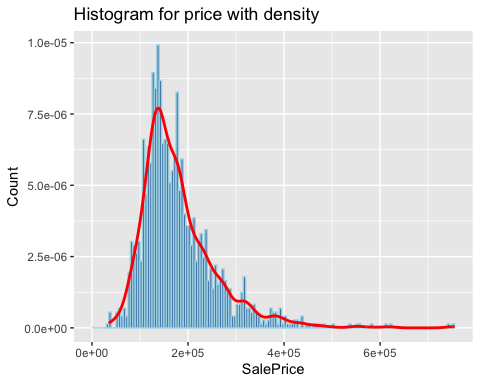
ks.test(imputed\_data$SalePrice, 'pnorm', mean(imputed\_data$SalePrice), sd(imputed\_data$SalePrice))

## 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, SalePrice: 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$quanti  
res.con$quali

# 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)[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) # better  
summary(m15)   
as.matrix(AIC(m14, m15))  
  
# delete garagearea bv very correlated to garage  
display(m16<- lm(log(SalePrice) ~OverallQual+log(GrLivArea)+GarageCars+TotalBsmtSF+YearBuilt+YearRemodAdd+Fireplaces+BsmtFinSF1+LotFrontage+WoodDeckSF+OpenPorchSF, data = train\_new))  
anova(m15,m16) # not better  
summary(m16)   
as.matrix(AIC(m14, m16))  
  
display(m17<- lm(log(SalePrice) ~OverallQual+log(GrLivArea)+GarageCars+TotalBsmtSF+YearBuilt+YearRemodAdd+Fireplaces+BsmtFinSF1+LotFrontage+WoodDeckSF+X2ndFlrSF, data = train\_new))  
anova(m15,m17) # not better   
summary(m17)   
as.matrix(AIC(m15, m17))  
vif(m17)  
  
display(m18<- lm(log(SalePrice) ~OverallQual+log(GrLivArea)+GarageCars+TotalBsmtSF+YearBuilt+YearRemodAdd+Fireplaces+BsmtFinSF1+LotFrontage+WoodDeckSF+HalfBath, data = train\_new))  
anova(m15,m18) # not better   
summary(m18)   
as.matrix(AIC(m15, m18))  
vif(m18)

We stop here, the model we end up with is :

display(m15<- lm(log(SalePrice) ~OverallQual+log(GrLivArea)+GarageCars+GarageArea+TotalBsmtSF+YearBuilt+YearRemodAdd+Fireplaces+BsmtFinSF1+LotFrontage+WoodDeckSF, data = train\_new))

## lm(formula = log(SalePrice) ~ OverallQual + log(GrLivArea) +   
## GarageCars + GarageArea + TotalBsmtSF + YearBuilt + YearRemodAdd +   
## Fireplaces + BsmtFinSF1 + LotFrontage + WoodDeckSF, data = train\_new)  
## coef.est coef.se  
## (Intercept) 0.13 0.55   
## OverallQual 0.08 0.01   
## log(GrLivArea) 0.36 0.02   
## GarageCars 0.02 0.01   
## GarageArea 0.00 0.00   
## TotalBsmtSF 0.00 0.00   
## YearBuilt 0.00 0.00   
## YearRemodAdd 0.00 0.00   
## Fireplaces 0.05 0.01   
## BsmtFinSF1 0.00 0.00   
## LotFrontage 0.00 0.00   
## WoodDeckSF 0.00 0.00   
## ---  
## n = 1016, k = 12  
## residual sd = 0.14, R-Squared = 0.88

as.matrix(AIC(m1, m2, m3, m4, m5, m6, m7, m8, m9, m10, m11, m12, m13, m14, m15))

## df AIC  
## m1 3 -177.4676  
## m2 4 -462.3334  
## m3 5 -594.1448  
## m4 6 -617.5179  
## m5 7 -785.4945  
## m6 8 -785.6116  
## m7 7 -783.1480  
## m8 8 -785.1551  
## m9 8 -906.5419  
## m10 9 -960.7879  
## m11 10 -964.6526  
## m12 10 -1022.0880  
## m13 11 -1118.0725  
## m14 12 -1146.1286  
## m15 13 -1158.2450

summary(m15)

##   
## Call:  
## lm(formula = log(SalePrice) ~ OverallQual + log(GrLivArea) +   
## GarageCars + GarageArea + TotalBsmtSF + YearBuilt + YearRemodAdd +   
## Fireplaces + BsmtFinSF1 + LotFrontage + WoodDeckSF, data = train\_new)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.84420 -0.06591 0.00739 0.07399 0.46146   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.293e-01 5.507e-01 0.235 0.814408   
## OverallQual 8.310e-02 5.233e-03 15.882 < 2e-16 \*\*\*  
## log(GrLivArea) 3.616e-01 1.911e-02 18.923 < 2e-16 \*\*\*  
## GarageCars 2.467e-02 1.311e-02 1.881 0.060196 .   
## GarageArea 1.278e-04 4.503e-05 2.837 0.004643 \*\*   
## TotalBsmtSF 1.114e-04 1.441e-05 7.729 2.63e-14 \*\*\*  
## YearBuilt 1.575e-03 2.002e-04 7.866 9.44e-15 \*\*\*  
## YearRemodAdd 2.652e-03 2.738e-04 9.686 < 2e-16 \*\*\*  
## Fireplaces 4.617e-02 7.908e-03 5.839 7.10e-09 \*\*\*  
## BsmtFinSF1 1.129e-04 1.159e-05 9.738 < 2e-16 \*\*\*  
## LotFrontage 1.149e-03 2.055e-04 5.592 2.90e-08 \*\*\*  
## WoodDeckSF 1.364e-04 3.639e-05 3.748 0.000188 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1359 on 1004 degrees of freedom  
## Multiple R-squared: 0.8844, Adjusted R-squared: 0.8831   
## F-statistic: 698.4 on 11 and 1004 DF, p-value: < 2.2e-16

summary(m15)$r.squared

## [1] 0.8844141

# check if there are any correlated predictors  
treshold\_vif<-function(rsq){  
 vif<- 1/(1-rsq)  
 return(vif)  
}  
treshold\_vif(summary(m15)$r.squared)

## [1] 8.651573

vif(m15) # there are not

## OverallQual log(GrLivArea) GarageCars GarageArea TotalBsmtSF   
## 2.950626 2.131832 5.506932 5.185798 1.871082   
## YearBuilt YearRemodAdd Fireplaces BsmtFinSF1 LotFrontage   
## 2.010659 1.743277 1.450775 1.378562 1.220065   
## WoodDeckSF   
## 1.137980

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

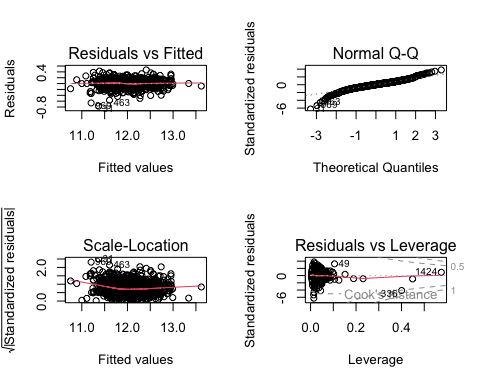
mod.fow <- stats::step(lm(SalePrice~. , data = train\_new), trace = FALSE, direction = "forward")  
summary(mod.fow)

##   
## 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   
## -91151 -10737 579 10346 168155   
##   
## Coefficients: (4 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.201e+06 1.237e+06 1.779 0.075578 .   
## Id 1.296e+00 1.881e+00 0.689 0.490782   
## LotFrontage 1.676e+01 4.784e+01 0.350 0.726194   
## LotArea 5.726e-01 1.441e-01 3.975 7.60e-05 \*\*\*  
## OverallQual 8.705e+03 1.182e+03 7.362 4.06e-13 \*\*\*  
## OverallCond 6.539e+03 9.322e+02 7.015 4.50e-12 \*\*\*  
## YearBuilt 4.290e+02 9.713e+01 4.417 1.12e-05 \*\*\*  
## YearRemodAdd 9.410e+01 6.150e+01 1.530 0.126337   
## MasVnrArea 2.369e+01 5.622e+00 4.214 2.77e-05 \*\*\*  
## BsmtFinSF1 4.282e+01 5.893e+00 7.266 8.00e-13 \*\*\*  
## BsmtFinSF2 3.613e+01 8.447e+00 4.277 2.09e-05 \*\*\*  
## BsmtUnfSF 2.716e+01 5.399e+00 5.030 5.91e-07 \*\*\*  
## TotalBsmtSF NA NA NA NA   
## X1stFlrSF 4.336e+01 6.244e+00 6.944 7.26e-12 \*\*\*  
## X2ndFlrSF 7.695e+01 7.209e+00 10.675 < 2e-16 \*\*\*  
## LowQualFinSF 1.661e+01 2.001e+01 0.830 0.406645   
## GrLivArea NA NA NA NA   
## BsmtFullBath 1.478e+03 2.364e+03 0.625 0.531945   
## BsmtHalfBath 5.522e+03 4.414e+03 1.251 0.211243   
## FullBath 5.797e+03 2.644e+03 2.192 0.028607 \*   
## HalfBath 3.412e+03 2.493e+03 1.368 0.171525   
## BedroomAbvGr -5.264e+03 1.649e+03 -3.192 0.001460 \*\*   
## KitchenAbvGr -7.862e+03 6.669e+03 -1.179 0.238736   
## TotRmsAbvGrd 4.069e+02 1.167e+03 0.349 0.727438   
## Fireplaces 5.407e+03 3.043e+03 1.777 0.075938 .   
## GarageCars 6.756e+03 2.601e+03 2.598 0.009541 \*\*   
## GarageArea 7.958e+00 8.895e+00 0.895 0.371195   
## WoodDeckSF 1.721e+01 6.996e+00 2.460 0.014069 \*   
## OpenPorchSF 3.103e+01 1.800e+01 1.724 0.085090 .   
## EnclosedPorch -1.618e+01 3.084e+01 -0.525 0.600027   
## X3SsnPorch 4.326e+01 2.484e+01 1.742 0.081917 .   
## ScreenPorch 6.633e+01 1.902e+01 3.487 0.000512 \*\*\*  
## PoolArea 4.408e+02 4.018e+02 1.097 0.272911   
## MiscVal 5.860e+00 6.179e+00 0.948 0.343267   
## YrSold -1.587e+03 6.088e+02 -2.608 0.009269 \*\*   
## NeighborhoodBlueste -2.460e+03 2.649e+04 -0.093 0.926007   
## NeighborhoodBrDale -1.472e+04 1.272e+04 -1.157 0.247587   
## NeighborhoodBrkSide -1.456e+04 1.069e+04 -1.362 0.173655   
## NeighborhoodClearCr -3.279e+04 1.109e+04 -2.955 0.003206 \*\*   
## NeighborhoodCollgCr -2.317e+04 9.252e+03 -2.505 0.012432 \*   
## NeighborhoodCrawfor 3.377e+03 1.033e+04 0.327 0.743806   
## NeighborhoodEdwards -2.761e+04 9.918e+03 -2.784 0.005481 \*\*   
## NeighborhoodGilbert -2.344e+04 9.546e+03 -2.455 0.014274 \*   
## NeighborhoodIDOTRR -2.592e+04 1.134e+04 -2.286 0.022458 \*   
## NeighborhoodMeadowV -7.974e+03 1.287e+04 -0.620 0.535557   
## NeighborhoodMitchel -3.503e+04 1.021e+04 -3.431 0.000628 \*\*\*  
## NeighborhoodNAmes -2.506e+04 9.639e+03 -2.600 0.009479 \*\*   
## NeighborhoodNoRidge 3.096e+03 1.048e+04 0.295 0.767807   
## NeighborhoodNPkVill 3.967e+02 1.323e+04 0.030 0.976089   
## NeighborhoodNridgHt -5.941e+03 9.554e+03 -0.622 0.534203   
## NeighborhoodNWAmes -3.244e+04 9.871e+03 -3.286 0.001055 \*\*   
## NeighborhoodOldTown -2.961e+04 1.028e+04 -2.879 0.004081 \*\*   
## NeighborhoodSawyer -2.494e+04 9.985e+03 -2.498 0.012678 \*   
## NeighborhoodSawyerW -2.191e+04 9.737e+03 -2.250 0.024682 \*   
## NeighborhoodSomerst -8.909e+03 9.627e+03 -0.925 0.354981   
## NeighborhoodStoneBr 2.219e+04 1.034e+04 2.146 0.032132 \*   
## NeighborhoodSWISU -2.575e+04 1.162e+04 -2.215 0.026986 \*   
## NeighborhoodTimber -2.423e+04 1.011e+04 -2.397 0.016724 \*   
## NeighborhoodVeenker 3.111e+03 1.306e+04 0.238 0.811797   
## ExterQualFa -1.243e+04 1.117e+04 -1.112 0.266405   
## ExterQualGd -2.026e+04 5.443e+03 -3.723 0.000209 \*\*\*  
## ExterQualTA -2.085e+04 6.075e+03 -3.433 0.000625 \*\*\*  
## BsmtQualFa -2.822e+04 7.322e+03 -3.855 0.000124 \*\*\*  
## BsmtQualGd -2.889e+04 4.223e+03 -6.841 1.44e-11 \*\*\*  
## BsmtQualNo Basement -4.184e+02 1.199e+04 -0.035 0.972178   
## BsmtQualTA -2.632e+04 5.157e+03 -5.105 4.04e-07 \*\*\*  
## KitchenQualFa -1.709e+04 7.544e+03 -2.265 0.023727 \*   
## KitchenQualGd -1.440e+04 4.098e+03 -3.512 0.000466 \*\*\*  
## KitchenQualTA -1.606e+04 4.619e+03 -3.476 0.000533 \*\*\*  
## GarageFinishNo Garage 2.692e+04 1.436e+04 1.875 0.061123 .   
## GarageFinishRFn -2.354e+03 2.444e+03 -0.963 0.335575   
## GarageFinishUnf 9.415e+00 2.962e+03 0.003 0.997464   
## FireplaceQuFa -1.681e+04 8.163e+03 -2.059 0.039745 \*   
## FireplaceQuGd -1.059e+04 6.323e+03 -1.674 0.094461 .   
## FireplaceQuNo Fireplace -5.050e+03 7.669e+03 -0.659 0.510377   
## FireplaceQuPo -1.045e+04 9.075e+03 -1.152 0.249699   
## FireplaceQuTA -1.348e+04 6.581e+03 -2.048 0.040817 \*   
## FoundationCBlock 1.008e+03 3.739e+03 0.269 0.787642   
## FoundationPConc 3.730e+03 4.245e+03 0.879 0.379855   
## FoundationSlab -1.492e+03 1.136e+04 -0.131 0.895468   
## FoundationStone 1.654e+04 1.484e+04 1.114 0.265397   
## FoundationWood -2.460e+04 1.557e+04 -1.579 0.114574   
## GarageTypeAttchd 1.434e+04 1.315e+04 1.090 0.275882   
## GarageTypeBasment 2.461e+04 1.518e+04 1.621 0.105316   
## GarageTypeBuiltIn 2.038e+04 1.374e+04 1.484 0.138288   
## GarageTypeCarPort -5.006e+03 1.613e+04 -0.310 0.756374   
## GarageTypeDetchd 1.586e+04 1.313e+04 1.208 0.227444   
## GarageTypeNo Garage NA NA NA NA   
## MSSubClass30 2.911e+03 5.574e+03 0.522 0.601679   
## MSSubClass40 1.546e+04 2.549e+04 0.607 0.544238   
## MSSubClass45 -4.384e+02 9.784e+03 -0.045 0.964274   
## MSSubClass50 1.157e+04 6.822e+03 1.696 0.090239 .   
## MSSubClass60 4.654e+02 6.236e+03 0.075 0.940527   
## MSSubClass70 3.850e+03 7.919e+03 0.486 0.626916   
## MSSubClass75 1.281e+04 1.133e+04 1.131 0.258524   
## MSSubClass80 3.090e+03 4.841e+03 0.638 0.523345   
## MSSubClass85 3.163e+03 7.272e+03 0.435 0.663737   
## MSSubClass90 -1.182e+04 6.825e+03 -1.732 0.083614 .   
## MSSubClass120 -2.375e+04 4.829e+03 -4.920 1.03e-06 \*\*\*  
## MSSubClass160 -2.602e+04 8.447e+03 -3.080 0.002132 \*\*   
## MSSubClass180 -2.425e+04 1.247e+04 -1.944 0.052244 .   
## MSSubClass190 -2.848e+02 7.626e+03 -0.037 0.970213   
## BsmtFinType1BLQ 3.956e+03 3.233e+03 1.224 0.221402   
## BsmtFinType1GLQ 3.403e+03 3.012e+03 1.130 0.258878   
## BsmtFinType1LwQ -4.368e+03 4.443e+03 -0.983 0.325789   
## BsmtFinType1No Basement NA NA NA NA   
## BsmtFinType1Rec -3.387e+03 3.427e+03 -0.988 0.323273   
## BsmtFinType1Unf -2.042e+02 3.351e+03 -0.061 0.951430   
## univ\_outl\_count -4.064e+03 2.573e+03 -1.580 0.114537   
## f.LotArea.L 1.007e+04 3.878e+03 2.598 0.009531 \*\*   
## secondfloor1 -2.613e+04 7.230e+03 -3.614 0.000318 \*\*\*  
## OpenPorch\_binary1 1.143e+03 5.202e+03 0.220 0.826081   
## EnclosedPorch\_binary1 1.272e+04 7.297e+03 1.743 0.081751 .   
## HasPorch\_binary1 -3.922e+03 5.302e+03 -0.740 0.459684   
## Pool\_binary1 -2.600e+05 2.783e+05 -0.934 0.350467   
## ---  
## 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: 0.91   
## F-statistic: 94.32 on 110 and 905 DF, p-value: < 2.2e-16

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] 0.9042377

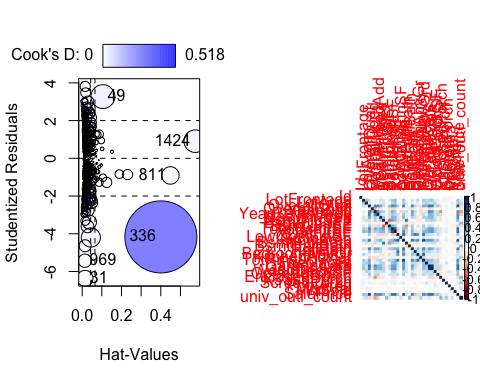
par(mfrow=c(2,2))  
plot(n00)



par(mfrow=c(1,2))  
influencePlot(n00)

## StudRes Hat CookD  
## 811 -0.9083526 0.44874073 0.03053530  
## 31 -6.3655632 0.01427277 0.02564896  
## 49 3.2718709 0.10553899 0.05685929  
## 336 -4.1656840 0.40034389 0.51807684  
## 969 -5.4297288 0.01346508 0.01778118  
## 1424 0.9083526 0.57499582 0.05074978

corrplot(descrCor, method = 'color')



# check if there are any correlated predictors  
treshold\_vif(summary(n00)$r.squared)

## [1] 10.44252

vif(n00) # there are not

## LotArea OverallQual OverallCond YearBuilt MasVnrArea BsmtFinSF1   
## 1.203086 3.300226 1.252545 2.835555 1.434956 5.357485   
## BsmtFinSF2 BsmtUnfSF X1stFlrSF X2ndFlrSF BsmtFullBath FullBath   
## 1.532296 4.334664 4.481430 4.573359 1.979919 2.890267   
## HalfBath BedroomAbvGr KitchenAbvGr Fireplaces GarageCars OpenPorchSF   
## 2.246294 1.670414 1.287002 1.529903 1.967063 1.189253   
## X3SsnPorch ScreenPorch PoolArea   
## 1.014968 1.063379 1.019176

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)

Now we have 2 potential models, m15 created with mechanic stepwise and n0 with automatic stepwise, we shall check which is better.

as.matrix(AIC(m15,n0))

## df AIC  
## m15 13 -1158.245  
## n0 16 -1289.761

anova(m15,n0)

## Analysis of Variance Table  
##   
## Model 1: log(SalePrice) ~ OverallQual + log(GrLivArea) + GarageCars +   
## GarageArea + TotalBsmtSF + YearBuilt + YearRemodAdd + Fireplaces +   
## BsmtFinSF1 + LotFrontage + WoodDeckSF  
## Model 2: log(SalePrice) ~ LotArea + OverallQual + OverallCond + YearBuilt +   
## BsmtFinSF1 + BsmtFinSF2 + X1stFlrSF + X2ndFlrSF + BsmtFullBath +   
## KitchenAbvGr + Fireplaces + GarageCars + ScreenPorch + PoolArea  
## Res.Df RSS Df Sum of Sq F Pr(>F)   
## 1 1004 18.544   
## 2 1001 16.197 3 2.3475 48.361 < 2.2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

summary(n0)$r.squared

## [1] 0.8990461

summary(m15)$r.squared

## [1] 0.8844141

We will proceed with the n0 model. ## 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

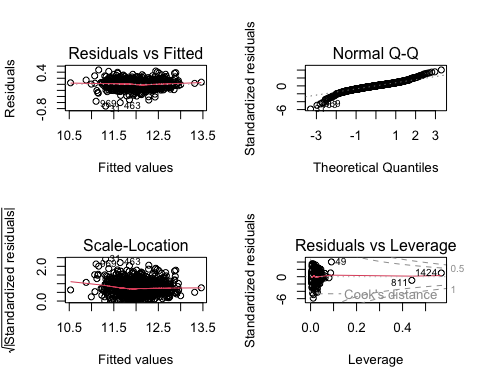
v1<- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + BsmtFinSF2+ log(X1stFlrSF)+ X2ndFlrSF +  
 BsmtFullBath + KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea, data = train\_new)  
  
  
# X2stFlrSF  
  
boxTidwell(log(SalePrice) ~ I(X2ndFlrSF+0.01),~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

We have been looking on the accuracy of the model, currently on 90% (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 also will consider interactions if needed.

#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] 0.9208465

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 12.699 26 2.5459 7.5179 < 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 -1484.929

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

## [1] 12.63369

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

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.883865 1 1.372540  
## log(LotArea) 2.993030 1 1.730038  
## OverallQual 3.644473 1 1.909050  
## OverallCond 1.345488 1 1.159952  
## YearBuilt 6.069508 1 2.463637  
## BsmtFinSF1 2.188038 1 1.479202  
## log(X1stFlrSF) 4.528370 1 2.127997  
## X2ndFlrSF 4.604548 1 2.145821  
## BsmtFullBath 1.897709 1 1.377574  
## FullBath 2.595230 1 1.610972  
## TotRmsAbvGrd 3.950502 1 1.987587  
## KitchenAbvGr 1.392456 1 1.180024  
## Fireplaces 1.684084 1 1.297722  
## GarageCars 2.137993 1 1.462188  
## ScreenPorch 1.092395 1 1.045177  
## PoolArea 1.029754 1 1.014768  
## Neighborhood 44.783008 24 1.082426

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] 0.9093023

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 14.551 5 1.6455 22.526 < 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 -1388.607

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

## [1] 11.02563

vif(b1)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.772856 1 1.331486  
## log(LotArea) 1.962296 1 1.400820  
## OverallQual 3.558811 1 1.886481  
## OverallCond 1.288881 1 1.135289  
## YearBuilt 2.578843 1 1.605878  
## BsmtFinSF1 2.064962 1 1.436998  
## log(X1stFlrSF) 4.055410 1 2.013805  
## X2ndFlrSF 4.041753 1 2.010411  
## BsmtFullBath 1.824323 1 1.350675  
## FullBath 2.355529 1 1.534773  
## TotRmsAbvGrd 3.749434 1 1.936345  
## KitchenAbvGr 1.319149 1 1.148542  
## Fireplaces 1.491125 1 1.221116  
## GarageCars 1.993455 1 1.411898  
## ScreenPorch 1.061447 1 1.030265  
## PoolArea 1.011505 1 1.005736  
## ExterQual 2.932361 3 1.196381

# 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] 0.9165639

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 14.551   
## 2 958 13.386 38 1.165 2.1941 5.333e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b1,b11)

## df AIC  
## b1 21 -1388.607  
## b11 59 -1397.393

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] 0.9139955

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 14.551   
## 2 992 13.798 4 0.75297 13.533 9.561e-11 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b1,b2)

## df AIC  
## b1 21 -1388.607  
## b2 25 -1434.590

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

## [1] 11.6273

vif(b2)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.784591 1 1.335886  
## log(LotArea) 1.975622 1 1.405568  
## OverallQual 3.827277 1 1.956343  
## OverallCond 1.310052 1 1.144575  
## YearBuilt 3.300162 1 1.816635  
## BsmtFinSF1 2.119687 1 1.455914  
## log(X1stFlrSF) 4.061162 1 2.015232  
## X2ndFlrSF 4.114987 1 2.028543  
## BsmtFullBath 1.838724 1 1.355996  
## FullBath 2.464208 1 1.569780  
## TotRmsAbvGrd 3.897475 1 1.974202  
## KitchenAbvGr 1.419794 1 1.191551  
## Fireplaces 1.495117 1 1.222750  
## GarageCars 2.024151 1 1.422727  
## ScreenPorch 1.064647 1 1.031818  
## PoolArea 1.011785 1 1.005875  
## ExterQual 3.810816 3 1.249788  
## BsmtQual 5.703513 4 1.243134

# 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] 0.9209938

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 13.798   
## 2 929 12.675 63 1.1228 1.3062 0.05931 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b2,b21)

## df AIC  
## b2 25 -1434.590  
## b21 88 -1394.821

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 13.386   
## 2 929 12.675 29 0.71072 1.7962 0.006334 \*\*  
## ---  
## 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] 0.9152059

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 13.798   
## 2 989 13.604 3 0.19419 4.7058 0.002872 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b2,b3)

## df AIC  
## b2 25 -1434.59  
## b3 28 -1442.99

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

## [1] 11.79328

vif(b3)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.795603 1 1.340001  
## log(LotArea) 1.978550 1 1.406609  
## OverallQual 3.910746 1 1.977561  
## OverallCond 1.378771 1 1.174211  
## YearBuilt 3.379908 1 1.838453  
## BsmtFinSF1 2.128593 1 1.458970  
## log(X1stFlrSF) 4.110490 1 2.027434  
## X2ndFlrSF 4.151802 1 2.037597  
## BsmtFullBath 1.847149 1 1.359099  
## FullBath 2.500402 1 1.581266  
## TotRmsAbvGrd 3.920177 1 1.979944  
## KitchenAbvGr 1.439679 1 1.199866  
## Fireplaces 1.496510 1 1.223319  
## GarageCars 2.065958 1 1.437344  
## ScreenPorch 1.068006 1 1.033444  
## PoolArea 1.014403 1 1.007176  
## ExterQual 5.792871 3 1.340136  
## BsmtQual 6.294666 4 1.258553  
## KitchenQual 4.879049 3 1.302334

#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] 0.9240861

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

## 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 13.604   
## 2 931 12.179 58 1.4247 1.8777 0.000121 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b3,b31)

## df AIC  
## b3 28 -1442.990  
## b31 86 -1439.386

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] 0.9154801

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 13.604   
## 2 986 13.560 3 0.04399 1.0662 0.3625

AIC(b3,b4)

## df AIC  
## b3 28 -1442.990  
## b4 31 -1440.281

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

## [1] 11.83154

vif(b4)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.821902 1 1.349778  
## log(LotArea) 1.981381 1 1.407615  
## OverallQual 3.924727 1 1.981092  
## OverallCond 1.382443 1 1.175774  
## YearBuilt 3.609870 1 1.899966  
## BsmtFinSF1 2.138896 1 1.462497  
## log(X1stFlrSF) 4.123073 1 2.030535  
## X2ndFlrSF 4.160924 1 2.039834  
## BsmtFullBath 1.851919 1 1.360852  
## FullBath 2.571444 1 1.603572  
## TotRmsAbvGrd 3.956873 1 1.989189  
## KitchenAbvGr 1.460846 1 1.208655  
## Fireplaces 1.524826 1 1.234838  
## GarageCars 3.039152 1 1.743317  
## ScreenPorch 1.070516 1 1.034657  
## PoolArea 1.015243 1 1.007593  
## ExterQual 5.994726 3 1.347809  
## BsmtQual 6.608662 4 1.266235  
## KitchenQual 4.971343 3 1.306408  
## GarageFinish 3.841405 3 1.251454

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] 0.9160358

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 13.604   
## 2 984 13.471 5 0.13315 1.9452 0.08442 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b3,b5)

## df AIC  
## b3 28 -1442.990  
## b5 33 -1442.983

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

## [1] 11.90984

vif(b5)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.805200 1 1.343577  
## log(LotArea) 1.993025 1 1.411745  
## OverallQual 3.968226 1 1.992041  
## OverallCond 1.381917 1 1.175550  
## YearBuilt 3.442269 1 1.855335  
## BsmtFinSF1 2.160387 1 1.469826  
## log(X1stFlrSF) 4.157099 1 2.038896  
## X2ndFlrSF 4.245480 1 2.060456  
## BsmtFullBath 1.859297 1 1.363560  
## FullBath 2.518636 1 1.587021  
## TotRmsAbvGrd 3.979396 1 1.994842  
## KitchenAbvGr 1.455279 1 1.206349  
## Fireplaces 5.981188 1 2.445647  
## GarageCars 2.071881 1 1.439403  
## ScreenPorch 1.077952 1 1.038245  
## PoolArea 1.032197 1 1.015971  
## ExterQual 5.877018 3 1.343361  
## BsmtQual 6.668181 4 1.267655  
## KitchenQual 5.070015 3 1.310695  
## FireplaceQu 8.894901 5 1.244269

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] 0.9167155

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 13.604   
## 2 984 13.362 5 0.24219 3.5671 0.003343 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b3,b6)

## df AIC  
## b3 28 -1442.990  
## b6 33 -1451.241

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

## [1] 12.00704

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

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.854522 1 1.361808  
## log(LotArea) 1.986207 1 1.409329  
## OverallQual 4.025380 1 2.006335  
## OverallCond 1.433331 1 1.197218  
## YearBuilt 5.499768 1 2.345158  
## BsmtFinSF1 2.164699 1 1.471292  
## log(X1stFlrSF) 4.172449 1 2.042657  
## X2ndFlrSF 4.213057 1 2.052573  
## BsmtFullBath 1.852836 1 1.361189  
## FullBath 2.533749 1 1.591775  
## TotRmsAbvGrd 3.938609 1 1.984593  
## KitchenAbvGr 1.441727 1 1.200719  
## Fireplaces 1.528964 1 1.236513  
## GarageCars 2.112501 1 1.453445  
## ScreenPorch 1.071556 1 1.035160  
## PoolArea 1.018893 1 1.009402  
## ExterQual 6.079982 3 1.350985  
## BsmtQual 19.299233 4 1.447746  
## KitchenQual 5.156699 3 1.314403  
## Foundation 17.285207 5 1.329742

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] 0.9163051

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 13.604   
## 2 983 13.428 6 0.17635 2.1517 0.04545 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b3,b7)

## df AIC  
## b3 28 -1442.990  
## b7 34 -1444.247

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

## [1] 11.94816

vif(b7)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.850012 1 1.360151  
## log(LotArea) 2.009293 1 1.417495  
## OverallQual 3.959592 1 1.989872  
## OverallCond 1.390136 1 1.179040  
## YearBuilt 3.980499 1 1.995119  
## BsmtFinSF1 2.146066 1 1.464946  
## log(X1stFlrSF) 4.207949 1 2.051329  
## X2ndFlrSF 4.363403 1 2.088876  
## BsmtFullBath 1.851925 1 1.360855  
## FullBath 2.545573 1 1.595485  
## TotRmsAbvGrd 3.959467 1 1.989841  
## KitchenAbvGr 1.507176 1 1.227671  
## Fireplaces 1.531266 1 1.237443  
## GarageCars 3.223722 1 1.795473  
## ScreenPorch 1.073660 1 1.036176  
## PoolArea 1.015149 1 1.007546  
## ExterQual 6.008461 3 1.348323  
## BsmtQual 6.668567 4 1.267664  
## KitchenQual 5.086397 3 1.311400  
## GarageType 4.860561 6 1.140838

# 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] 0.9272418

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 13.428   
## 2 919 11.673 64 1.7546 2.1584 9.544e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b7,b71)

## df AIC  
## b7 34 -1444.247  
## b71 98 -1458.524

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

## [1] 13.74416

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] 0.9177902

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 13.428   
## 2 969 13.189 14 0.23825 1.2503 0.2327

AIC(b7,b8)

## df AIC  
## b7 34 -1444.247  
## b8 48 -1434.436

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

## [1] 12.16399

vif(b8) # highly correlated

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 2.000031 1 1.414224  
## log(LotArea) 2.854043 1 1.689391  
## OverallQual 4.199898 1 2.049365  
## OverallCond 1.450141 1 1.204218  
## YearBuilt 8.553317 1 2.924605  
## BsmtFinSF1 2.191694 1 1.480437  
## log(X1stFlrSF) 4.865744 1 2.205843  
## X2ndFlrSF 9.267543 1 3.044264  
## BsmtFullBath 1.925040 1 1.387458  
## FullBath 2.600896 1 1.612729  
## TotRmsAbvGrd 4.210533 1 2.051958  
## KitchenAbvGr 2.962995 1 1.721335  
## Fireplaces 1.607317 1 1.267800  
## GarageCars 3.304628 1 1.817863  
## ScreenPorch 1.084370 1 1.041331  
## PoolArea 1.032979 1 1.016356  
## ExterQual 6.392847 3 1.362330  
## BsmtQual 8.307030 4 1.302959  
## KitchenQual 5.440925 3 1.326209  
## GarageType 7.035398 6 1.176542  
## MSSubClass 412.372468 14 1.239947

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] 0.9167085

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 13.428   
## 2 978 13.363 5 0.064712 0.9472 0.4495

AIC(b7,b9)

## df AIC  
## b7 34 -1444.247  
## b9 39 -1439.155

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

## [1] 12.00602

#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] 0.916316

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 13.428   
## 2 982 13.426 1 0.0017531 0.1282 0.7204

AIC(b7,b10)

## df AIC  
## b7 34 -1444.247  
## b10 35 -1442.379

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

## [1] 11.94972

vif(b10)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.851241 1 1.360603  
## log(LotArea) 2.365393 1 1.537983  
## OverallQual 3.977146 1 1.994278  
## OverallCond 1.390397 1 1.179151  
## YearBuilt 3.981444 1 1.995356  
## BsmtFinSF1 2.154644 1 1.467871  
## log(X1stFlrSF) 4.213213 1 2.052611  
## X2ndFlrSF 4.388753 1 2.094935  
## BsmtFullBath 1.856241 1 1.362439  
## FullBath 2.561100 1 1.600344  
## TotRmsAbvGrd 3.993347 1 1.998336  
## KitchenAbvGr 1.507783 1 1.227918  
## Fireplaces 1.538267 1 1.240269  
## GarageCars 3.224856 1 1.795789  
## ScreenPorch 1.073693 1 1.036192  
## PoolArea 1.025942 1 1.012888  
## ExterQual 6.024744 3 1.348931  
## BsmtQual 6.681612 4 1.267973  
## KitchenQual 5.109752 3 1.312401  
## GarageType 4.888545 6 1.141384  
## f.LotArea 1.381843 1 1.175518

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] 0.9165639

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 13.428   
## 2 958 13.386 25 0.041515 0.1188 1

AIC(b7,b11)

## df AIC  
## b7 34 -1444.247  
## b11 59 -1397.393

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

## df AIC  
## b7 34 -1444.247  
## b71 98 -1458.524  
## b3 28 -1442.990  
## b31 86 -1439.386  
## b2 25 -1434.590  
## b21 88 -1394.821  
## b1 21 -1388.607  
## b11 59 -1397.393

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 21653.44  
## 2 Model with test\_new sample 0.92 23875.29

## 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

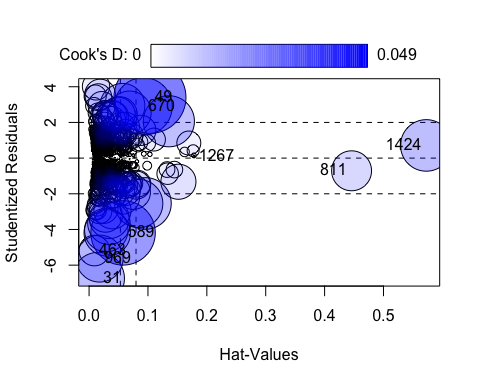
## Study the presence of a posteriori influential values

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  
## 670 2.89628603 0.092274834 3.134842e-02  
## 811 -0.71067259 0.445813373 1.505531e-02  
## 31 -6.74281296 0.015733639 2.575937e-02  
## 49 3.40357328 0.103391389 4.895136e-02  
## 589 -4.19105723 0.057716465 3.919112e-02  
## 1267 0.08606811 0.178252854 5.957378e-05  
## 969 -5.63304305 0.017933428 2.081399e-02  
## 1424 0.71067259 0.572738924 2.508743e-02  
## 463 -5.18072447 0.009054112 8.851399e-03

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

## [1] 25

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

## [1] 28

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

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

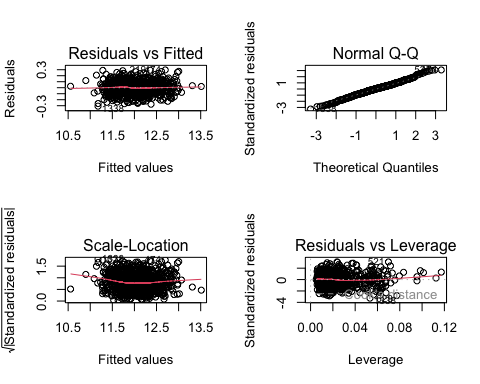
train\_new<-train\_new[-llremreg,]  
  
final\_modelp <- lm(log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + log(X1stFlrSF)+ X2ndFlrSF +  
 BsmtFullBath + FullBath + TotRmsAbvGrd+  
 KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea +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   
## -0.275405 -0.060107 0.002103 0.058319 0.297301   
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.431e+00 4.593e-01 3.116 0.001889 \*\*   
## LotFrontage -2.708e-04 1.821e-04 -1.487 0.137245   
## log(LotArea) 1.028e-01 8.595e-03 11.963 < 2e-16 \*\*\*  
## OverallQual 6.059e-02 4.351e-03 13.925 < 2e-16 \*\*\*  
## OverallCond 5.261e-02 3.199e-03 16.443 < 2e-16 \*\*\*  
## YearBuilt 3.118e-03 2.035e-04 15.322 < 2e-16 \*\*\*  
## BsmtFinSF1 7.554e-05 1.029e-05 7.343 4.46e-13 \*\*\*  
## log(X1stFlrSF) 4.004e-01 2.030e-02 19.727 < 2e-16 \*\*\*  
## X2ndFlrSF 2.439e-04 1.482e-05 16.463 < 2e-16 \*\*\*  
## BsmtFullBath 2.171e-02 8.004e-03 2.713 0.006787 \*\*   
## FullBath 2.980e-02 8.811e-03 3.382 0.000748 \*\*\*  
## TotRmsAbvGrd -9.009e-04 3.822e-03 -0.236 0.813686   
## KitchenAbvGr -7.925e-02 1.792e-02 -4.422 1.09e-05 \*\*\*  
## Fireplaces 2.084e-02 5.799e-03 3.594 0.000342 \*\*\*  
## GarageCars 4.574e-02 7.177e-03 6.373 2.88e-10 \*\*\*  
## ScreenPorch 2.876e-04 6.043e-05 4.759 2.24e-06 \*\*\*  
## PoolArea NA NA NA NA   
## ExterQualFa -4.796e-02 4.408e-02 -1.088 0.276827   
## ExterQualGd -7.284e-03 2.118e-02 -0.344 0.730963   
## ExterQualTA -4.701e-02 2.367e-02 -1.987 0.047261 \*   
## BsmtQualFa -1.715e-01 2.823e-02 -6.077 1.77e-09 \*\*\*  
## BsmtQualGd -9.343e-02 1.575e-02 -5.931 4.20e-09 \*\*\*  
## BsmtQualNo Basement -2.196e-01 2.872e-02 -7.647 4.99e-14 \*\*\*  
## BsmtQualTA -1.009e-01 1.856e-02 -5.439 6.79e-08 \*\*\*  
## KitchenQualFa -1.191e-01 2.976e-02 -4.002 6.77e-05 \*\*\*  
## KitchenQualGd -6.566e-02 1.581e-02 -4.152 3.59e-05 \*\*\*  
## KitchenQualTA -9.309e-02 1.773e-02 -5.250 1.87e-07 \*\*\*  
## GarageTypeAttchd 8.343e-02 5.053e-02 1.651 0.099042 .   
## GarageTypeBasment 1.190e-01 5.787e-02 2.056 0.040074 \*   
## GarageTypeBuiltIn 8.791e-02 5.242e-02 1.677 0.093892 .   
## GarageTypeCarPort 3.399e-03 6.392e-02 0.053 0.957600   
## GarageTypeDetchd 9.628e-02 5.029e-02 1.914 0.055867 .   
## GarageTypeNo Garage 9.469e-02 5.425e-02 1.746 0.081195 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.0961 on 957 degrees of freedom  
## Multiple R-squared: 0.9407, Adjusted R-squared: 0.9387   
## F-statistic: 489.3 on 31 and 957 DF, p-value: < 2.2e-16

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

## Start: AIC=-4445  
## 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=-4445  
## 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 0.0995 8.9380 -4475.3  
## - ExterQual 3 0.1178 8.9563 -4452.6  
## - TotRmsAbvGrd 1 0.0005 8.8389 -4451.8  
## - LotFrontage 1 0.0204 8.8589 -4449.6  
## <none> 8.8384 -4445.0  
## - BsmtFullBath 1 0.0680 8.9064 -4444.3  
## - FullBath 1 0.1056 8.9441 -4440.1  
## - Fireplaces 1 0.1193 8.9577 -4438.6  
## - KitchenQual 3 0.2679 9.1064 -4436.2  
## - KitchenAbvGr 1 0.1806 9.0190 -4431.9  
## - ScreenPorch 1 0.2092 9.0476 -4428.8  
## - GarageCars 1 0.3751 9.2135 -4410.8  
## - BsmtQual 4 0.6570 9.4954 -4401.7  
## - BsmtFinSF1 1 0.4980 9.3364 -4397.7  
## - log(LotArea) 1 1.3218 10.1602 -4314.1  
## - OverallQual 1 1.7907 10.6291 -4269.4  
## - YearBuilt 1 2.1682 11.0066 -4234.9  
## - OverallCond 1 2.4970 11.3354 -4205.8  
## - X2ndFlrSF 1 2.5030 11.3414 -4205.3  
## - log(X1stFlrSF) 1 3.5940 12.4324 -4114.5  
##   
## Step: AIC=-4475.3  
## 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 0.1142 9.0521 -4483.4  
## - TotRmsAbvGrd 1 0.0001 8.9380 -4482.2  
## - LotFrontage 1 0.0222 8.9602 -4479.7  
## <none> 8.9380 -4475.3  
## - BsmtFullBath 1 0.0687 9.0067 -4474.6  
## - FullBath 1 0.1019 9.0399 -4471.0  
## - Fireplaces 1 0.1228 9.0608 -4468.7  
## - KitchenQual 3 0.2593 9.1973 -4467.7  
## - ScreenPorch 1 0.2098 9.1478 -4459.2  
## - KitchenAbvGr 1 0.2345 9.1724 -4456.6  
## - BsmtQual 4 0.6712 9.6092 -4431.3  
## - BsmtFinSF1 1 0.4868 9.4248 -4429.8  
## - GarageCars 1 0.5232 9.4612 -4425.9  
## - log(LotArea) 1 1.2763 10.2143 -4350.2  
## - OverallQual 1 1.8746 10.8126 -4293.9  
## - YearBuilt 1 2.4477 11.3857 -4242.8  
## - OverallCond 1 2.5317 11.4696 -4235.5  
## - X2ndFlrSF 1 2.5976 11.5356 -4229.9  
## - log(X1stFlrSF) 1 3.6313 12.5692 -4145.0  
##   
## Step: AIC=-4483.44  
## 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 0.0003 9.0525 -4490.3  
## - LotFrontage 1 0.0262 9.0783 -4487.5  
## <none> 9.0521 -4483.4  
## - BsmtFullBath 1 0.0746 9.1267 -4482.2  
## - FullBath 1 0.0996 9.1517 -4479.5  
## - Fireplaces 1 0.1100 9.1621 -4478.4  
## - ScreenPorch 1 0.2069 9.2591 -4468.0  
## - KitchenAbvGr 1 0.2271 9.2792 -4465.8  
## - KitchenQual 3 0.4618 9.5140 -4454.9  
## - BsmtFinSF1 1 0.4539 9.5060 -4442.0  
## - BsmtQual 4 0.7181 9.7703 -4435.5  
## - GarageCars 1 0.5262 9.5783 -4434.5  
## - log(LotArea) 1 1.2732 10.3253 -4360.2  
## - OverallQual 1 2.3980 11.4502 -4257.9  
## - OverallCond 1 2.5749 11.6270 -4242.8  
## - X2ndFlrSF 1 2.6383 11.6904 -4237.4  
## - YearBuilt 1 2.6748 11.7269 -4234.3  
## - log(X1stFlrSF) 1 3.7191 12.7712 -4149.9  
##   
## Step: AIC=-4490.3  
## 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 0.0266 9.0791 -4494.3  
## <none> 9.0525 -4490.3  
## - BsmtFullBath 1 0.0745 9.1269 -4489.1  
## - FullBath 1 0.0996 9.1520 -4486.4  
## - Fireplaces 1 0.1100 9.1624 -4485.3  
## - ScreenPorch 1 0.2080 9.2605 -4474.7  
## - KitchenAbvGr 1 0.2438 9.2962 -4470.9  
## - KitchenQual 3 0.4617 9.5142 -4461.8  
## - BsmtFinSF1 1 0.4682 9.5207 -4447.3  
## - BsmtQual 4 0.7247 9.7772 -4441.7  
## - GarageCars 1 0.5259 9.5784 -4441.3  
## - log(LotArea) 1 1.2751 10.3276 -4366.9  
## - OverallQual 1 2.4006 11.4531 -4264.6  
## - OverallCond 1 2.5750 11.6275 -4249.6  
## - YearBuilt 1 2.6764 11.7288 -4241.0  
## - X2ndFlrSF 1 4.8751 13.9275 -4071.1  
## - log(X1stFlrSF) 1 4.9125 13.9649 -4068.5  
##   
## Step: AIC=-4494.29  
## 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> 9.0791 -4494.3  
## - BsmtFullBath 1 0.0756 9.1547 -4493.0  
## - FullBath 1 0.1017 9.1808 -4490.2  
## - Fireplaces 1 0.1145 9.1936 -4488.8  
## - ScreenPorch 1 0.2049 9.2840 -4479.1  
## - KitchenAbvGr 1 0.2399 9.3190 -4475.4  
## - KitchenQual 3 0.4658 9.5449 -4465.5  
## - BsmtFinSF1 1 0.4753 9.5544 -4450.7  
## - BsmtQual 4 0.7109 9.7900 -4447.3  
## - GarageCars 1 0.5118 9.5909 -4447.0  
## - log(LotArea) 1 1.4840 10.5631 -4351.5  
## - OverallQual 1 2.4188 11.4979 -4267.6  
## - OverallCond 1 2.5626 11.6417 -4255.3  
## - YearBuilt 1 2.6524 11.7315 -4247.7  
## - X2ndFlrSF 1 4.8488 13.9279 -4078.0  
## - log(X1stFlrSF) 1 4.9122 13.9913 -4073.5

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) 1.695567e+00  
## log(LotArea) log(LotArea) 9.760302e-02  
## OverallQual OverallQual 6.652039e-02  
## OverallCond OverallCond 5.278278e-02  
## YearBuilt YearBuilt 3.054057e-03  
## BsmtFinSF1 BsmtFinSF1 7.796773e-05  
## log(X1stFlrSF) log(X1stFlrSF) 3.917527e-01  
## X2ndFlrSF X2ndFlrSF 2.331492e-04  
## BsmtFullBath BsmtFullBath 2.316898e-02  
## FullBath FullBath 2.829100e-02  
## KitchenAbvGr KitchenAbvGr -7.036863e-02  
## Fireplaces Fireplaces 2.419235e-02  
## GarageCars GarageCars 4.925005e-02  
## ScreenPorch ScreenPorch 2.979136e-04  
## BsmtQualFa BsmtQualFa -1.558118e-01  
## BsmtQualGd BsmtQualGd -9.744856e-02  
## BsmtQualNo Basement BsmtQualNo Basement -2.208620e-01  
## BsmtQualTA BsmtQualTA -1.069885e-01  
## KitchenQualFa KitchenQualFa -6.092359e-02  
## KitchenQualGd KitchenQualGd -4.293402e-02  
## KitchenQualTA KitchenQualTA -7.438076e-02  
## Filtered\_coefficient  
## (Intercept) 1.581052e+00  
## log(LotArea) 9.267094e-02  
## OverallQual 6.641053e-02  
## OverallCond 5.203768e-02  
## YearBuilt 3.122607e-03  
## BsmtFinSF1 7.241134e-05  
## log(X1stFlrSF) 3.947548e-01  
## X2ndFlrSF 2.410743e-04  
## BsmtFullBath 2.280715e-02  
## FullBath 2.876738e-02  
## KitchenAbvGr -8.633717e-02  
## Fireplaces 2.011641e-02  
## GarageCars 4.294882e-02  
## ScreenPorch 2.833382e-04  
## BsmtQualFa -1.759836e-01  
## BsmtQualGd -9.480145e-02  
## BsmtQualNo Basement -2.228569e-01  
## BsmtQualTA -1.074626e-01  
## KitchenQualFa -1.282237e-01  
## KitchenQualGd -6.593118e-02  
## KitchenQualTA -1.094106e-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 19421.9  
## 2 Model with test\_new sample 0.92 23845.2

# Test predictions

## Test standarization

test<-read.delim("test.csv", sep=',')   
test$Alley[which(is.na(test$Alley))] <- "No alley access"  
test$BsmtQual[which(is.na(test$BsmtQual))] <- "No Basement"  
test$BsmtCond[which(is.na(test$BsmtCond))] <- "No Basement"  
test$BsmtExposure[which(is.na(test$BsmtExposure))] <- "No Basement"  
test$BsmtFinType1[which(is.na(test$BsmtFinType1))] <- "No Basement"  
test$BsmtFinType2[which(is.na(test$BsmtFinType2))] <- "No Basement"  
test$FireplaceQu[which(is.na(test$FireplaceQu))] <- "No Fireplace"  
test$GarageType[which(is.na(test$GarageType))] <- "No Garage"  
test$GarageFinish[which(is.na(test$GarageFinish))] <- "No Garage"  
test$GarageCond[which(is.na(test$GarageCond))] <- "No Garage"  
test$PoolQC[which(is.na(test$PoolQC))] <- "No Pool"  
test$Fence[which(is.na(test$Fence))] <- "No Fence"  
test$MiscFeature[which(is.na(test$MiscFeature))] <- "None"  
char\_cols <- which(sapply(test, is.character))  
test[, char\_cols] <- lapply(test[, char\_cols], as.factor)   
test$MSSubClass <- factor(test$MSSubClass)  
test$MoSold <- factor(test$MoSold)  
vect<- c("Neighborhood", "ExterQual", "BsmtQual", "KitchenQual", "GarageFinish", "FireplaceQu", "Foundation", "GarageType", "MSSubClass", "BsmtFinType1")  
factor\_test<-test[,c(vect)] # select the first 10 factor variables that are more related with the target  
num\_cols <- names(test)[sapply(test, is.numeric)]  
test2 <- test[,c(num\_cols, vect)]  
  
## NEW FEATURES  
# - EnclosedPorch\_binary: 0 = NO enclosed porch area / 1 = they have  
test2$EnclosedPorch\_binary<-0  
test2$EnclosedPorch\_binary[which(test2$EnclosedPorch!=0)]<-1  
test2$EnclosedPorch\_binary<-as.factor(test2$EnclosedPorch\_binary)  
  
# - OpenPorch\_binary: 0 = NO open porch area / 1 = they have  
test2$OpenPorch\_binary<-0  
test2$OpenPorch\_binary[which(test2$OpenPorchSF!=0)]<-1  
test2$OpenPorch\_binary<-as.factor(test2$OpenPorch\_binary)  
  
# - HasPorch\_binary: 0 = NO porch area / 1 = they have (mixing both binary variables)  
test2$HasPorch\_binary<-0  
test2$HasPorch\_binary[which(test2$EnclosedPorch!=0 | test2$OpenPorchSF!=0)]<-1  
test2$HasPorch\_binary<-as.factor(test2$HasPorch\_binary)  
  
# - Pool\_binary: 0 = no pool / 1 = pool  
test2$Pool\_binary<-0  
test2$Pool\_binary[which(test2$PoolArea!=0)]<-1  
test2$Pool\_binary<-as.factor(test2$Pool\_binary)  
  
# - secondfloor: 0 = not second floor / 1 = has second floor   
test2$secondfloor<-0  
test2$secondfloor[which(test2$X2ndFlrSF!=0)]<-1  
test2$secondfloor<-factor(test2$secondfloor)  
  
test2$univ\_outl\_count <- 0 # initialize  
  
  
columns\_to\_exclude <- c("Id","univ\_outl\_count")  
  
# apply the function to each column  
for (column\_name in colnames(test2)) {  
 if (!(column\_name %in% columns\_to\_exclude) && is.numeric(test2[[column\_name]])) {  
 test2 <- update\_outliers\_count(test2, column\_name)  
 }  
}  
  
mice\_imp<-mice(test2[, !names(test2) %in% "GarageYrBlt"],method = "cart")

##   
## iter imp variable  
## 1 1 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 1 2 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 1 3 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 1 4 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 1 5 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 2 1 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 2 2 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 2 3 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 2 4 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 2 5 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 3 1 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 3 2 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 3 3 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 3 4 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 3 5 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 4 1 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 4 2 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 4 3 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 4 4 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 4 5 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 5 1 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 5 2 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 5 3 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 5 4 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual  
## 5 5 LotFrontage MasVnrArea BsmtFinSF1 BsmtFinSF2 BsmtUnfSF TotalBsmtSF BsmtFullBath BsmtHalfBath GarageCars GarageArea KitchenQual

## Warning: Number of logged events: 275

imputed\_test<-complete(mice\_imp)  
  
predictions <- predict(final\_model, imputed\_test)  
predictions <- exp(predictions)  
prediccion <- data.frame("id"=imputed\_test$Id, "SalePrice"=predictions)  
prediccion

## id SalePrice  
## 1 1461 123388.69  
## 2 1462 160425.98  
## 3 1463 175850.60  
## 4 1464 192368.63  
## 5 1465 194348.93  
## 6 1466 163687.43  
## 7 1467 183756.37  
## 8 1468 158179.67  
## 9 1469 189088.95  
## 10 1470 115882.29  
## 11 1471 206343.48  
## 12 1472 90872.48  
## 13 1473 90144.65  
## 14 1474 149966.21  
## 15 1475 117271.56  
## 16 1476 360712.96  
## 17 1477 233423.30  
## 18 1478 301333.35  
## 19 1479 295015.01  
## 20 1480 430265.02  
## 21 1481 296240.92  
## 22 1482 207876.30  
## 23 1483 187007.88  
## 24 1484 163433.14  
## 25 1485 194262.88  
## 26 1486 198316.99  
## 27 1487 357246.07  
## 28 1488 232942.19  
## 29 1489 183563.81  
## 30 1490 255060.14  
## 31 1491 195184.24  
## 32 1492 99493.65  
## 33 1493 190952.99  
## 34 1494 284204.95  
## 35 1495 292519.83  
## 36 1496 241048.45  
## 37 1497 176708.25  
## 38 1498 148903.65  
## 39 1499 150332.09  
## 40 1500 142365.48  
## 41 1501 160909.96  
## 42 1502 135793.45  
## 43 1503 279778.51  
## 44 1504 230563.08  
## 45 1505 215661.01  
## 46 1506 187370.77  
## 47 1507 250751.53  
## 48 1508 208713.71  
## 49 1509 160371.99  
## 50 1510 139550.50  
## 51 1511 143709.21  
## 52 1512 184282.21  
## 53 1513 160201.79  
## 54 1514 160882.83  
## 55 1515 219028.96  
## 56 1516 163528.55  
## 57 1517 176441.47  
## 58 1518 132534.30  
## 59 1519 217739.31  
## 60 1520 125904.82  
## 61 1521 129397.12  
## 62 1522 178427.98  
## 63 1523 107041.99  
## 64 1524 131338.80  
## 65 1525 119763.28  
## 66 1526 121910.01  
## 67 1527 105510.22  
## 68 1528 143574.17  
## 69 1529 134267.59  
## 70 1530 190321.00  
## 71 1531 109320.76  
## 72 1532 104329.90  
## 73 1533 149170.77  
## 74 1534 126907.02  
## 75 1535 159719.15  
## 76 1536 111585.45  
## 77 1537 66591.95  
## 78 1538 173717.35  
## 79 1539 210514.45  
## 80 1540 106832.03  
## 81 1541 133237.60  
## 82 1542 134874.08  
## 83 1543 177679.61  
## 84 1544 95237.26  
## 85 1545 118058.82  
## 86 1546 130381.43  
## 87 1547 138345.21  
## 88 1548 132588.87  
## 89 1549 105925.41  
## 90 1550 126208.92  
## 91 1551 113499.51  
## 92 1552 118746.35  
## 93 1553 144643.08  
## 94 1554 110684.23  
## 95 1555 162983.84  
## 96 1556 88912.79  
## 97 1557 108998.74  
## 98 1558 92874.50  
## 99 1559 80837.41  
## 100 1560 132495.07  
## 101 1561 122734.64  
## 102 1562 127704.62  
## 103 1563 115415.16  
## 104 1564 164875.08  
## 105 1565 135152.01  
## 106 1566 234579.68  
## 107 1567 79316.28  
## 108 1568 215238.09  
## 109 1569 114878.25  
## 110 1570 147129.90  
## 111 1571 117620.81  
## 112 1572 142565.47  
## 113 1573 251080.01  
## 114 1574 122621.60  
## 115 1575 238042.92  
## 116 1576 251256.87  
## 117 1577 182667.28  
## 118 1578 132975.05  
## 119 1579 144161.83  
## 120 1580 189251.75  
## 121 1581 159627.49  
## 122 1582 126395.40  
## 123 1583 297224.94  
## 124 1584 219933.99  
## 125 1585 146148.95  
## 126 1586 73202.42  
## 127 1587 108931.31  
## 128 1588 146538.71  
## 129 1589 112221.78  
## 130 1590 143460.58  
## 131 1591 93751.02  
## 132 1592 118024.61  
## 133 1593 129704.38  
## 134 1594 131676.71  
## 135 1595 100740.37  
## 136 1596 197314.41  
## 137 1597 165096.47  
## 138 1598 215032.08  
## 139 1599 185425.87  
## 140 1600 177319.59  
## 141 1601 69610.26  
## 142 1602 111426.47  
## 143 1603 91828.54  
## 144 1604 273817.17  
## 145 1605 257725.30  
## 146 1606 185388.86  
## 147 1607 175687.23  
## 148 1608 216511.89  
## 149 1609 185786.15  
## 150 1610 147104.27  
## 151 1611 142862.48  
## 152 1612 181610.84  
## 153 1613 158502.81  
## 154 1614 124727.64  
## 155 1615 87919.69  
## 156 1616 75868.36  
## 157 1617 91133.99  
## 158 1618 123338.91  
## 159 1619 145295.78  
## 160 1620 153763.51  
## 161 1621 131987.98  
## 162 1622 134857.28  
## 163 1623 264030.40  
## 164 1624 218498.48  
## 165 1625 114882.59  
## 166 1626 173568.06  
## 167 1627 181859.85  
## 168 1628 281259.33  
## 169 1629 185319.06  
## 170 1630 346884.18  
## 171 1631 200127.67  
## 172 1632 234063.95  
## 173 1633 173205.30  
## 174 1634 185050.48  
## 175 1635 171857.27  
## 176 1636 159328.22  
## 177 1637 194228.78  
## 178 1638 218626.10  
## 179 1639 186702.66  
## 180 1640 286222.14  
## 181 1641 185502.87  
## 182 1642 231804.91  
## 183 1643 220953.39  
## 184 1644 222041.52  
## 185 1645 235425.81  
## 186 1646 166809.00  
## 187 1647 167786.63  
## 188 1648 126845.21  
## 189 1649 138888.57  
## 190 1650 118839.23  
## 191 1651 121091.66  
## 192 1652 94290.00  
## 193 1653 97991.09  
## 194 1654 138780.45  
## 195 1655 139561.72  
## 196 1656 139072.10  
## 197 1657 140110.70  
## 198 1658 142761.16  
## 199 1659 120524.64  
## 200 1660 154501.81  
## 201 1661 490901.76  
## 202 1662 419903.11  
## 203 1663 373645.46  
## 204 1664 508964.76  
## 205 1665 310575.17  
## 206 1666 328532.71  
## 207 1667 415063.27  
## 208 1668 341920.06  
## 209 1669 319173.11  
## 210 1670 357410.92  
## 211 1671 248948.79  
## 212 1672 427055.92  
## 213 1673 287040.39  
## 214 1674 236216.20  
## 215 1675 191555.09  
## 216 1676 189169.52  
## 217 1677 217064.78  
## 218 1678 467297.04  
## 219 1679 361315.94  
## 220 1680 324781.47  
## 221 1681 265636.83  
## 222 1682 305465.05  
## 223 1683 180634.72  
## 224 1684 184992.22  
## 225 1685 172332.94  
## 226 1686 171033.16  
## 227 1687 179533.17  
## 228 1688 193243.70  
## 229 1689 195236.63  
## 230 1690 200226.38  
## 231 1691 184325.94  
## 232 1692 290024.30  
## 233 1693 178952.96  
## 234 1694 176610.50  
## 235 1695 166273.44  
## 236 1696 286838.42  
## 237 1697 173697.39  
## 238 1698 346009.17  
## 239 1699 268712.66  
## 240 1700 236894.07  
## 241 1701 267402.19  
## 242 1702 238894.17  
## 243 1703 241162.33  
## 244 1704 274032.73  
## 245 1705 242187.83  
## 246 1706 439682.50  
## 247 1707 212333.51  
## 248 1708 193622.95  
## 249 1709 250804.39  
## 250 1710 210638.71  
## 251 1711 277419.13  
## 252 1712 234847.39  
## 253 1713 268391.96  
## 254 1714 216318.87  
## 255 1715 197311.92  
## 256 1716 188579.16  
## 257 1717 175569.07  
## 258 1718 140378.00  
## 259 1719 199788.84  
## 260 1720 233562.21  
## 261 1721 177574.24  
## 262 1722 136560.30  
## 263 1723 168220.82  
## 264 1724 203713.22  
## 265 1725 239680.77  
## 266 1726 191001.28  
## 267 1727 166516.67  
## 268 1728 180448.79  
## 269 1729 165749.05  
## 270 1730 160144.18  
## 271 1731 120995.99  
## 272 1732 124727.11  
## 273 1733 112403.23  
## 274 1734 122190.91  
## 275 1735 142942.09  
## 276 1736 112437.84  
## 277 1737 348710.94  
## 278 1738 246828.24  
## 279 1739 272582.28  
## 280 1740 188952.87  
## 281 1741 180256.11  
## 282 1742 161409.25  
## 283 1743 171656.15  
## 284 1744 323205.78  
## 285 1745 198492.15  
## 286 1746 240867.97  
## 287 1747 232104.03  
## 288 1748 246892.84  
## 289 1749 149438.63  
## 290 1750 144828.37  
## 291 1751 266908.14  
## 292 1752 116408.85  
## 293 1753 142346.48  
## 294 1754 214948.19  
## 295 1755 175757.75  
## 296 1756 126921.73  
## 297 1757 118857.41  
## 298 1758 143390.11  
## 299 1759 177025.04  
## 300 1760 157239.97  
## 301 1761 155256.86  
## 302 1762 170381.08  
## 303 1763 181231.03  
## 304 1764 111916.77  
## 305 1765 181152.95  
## 306 1766 199925.07  
## 307 1767 260578.93  
## 308 1768 144371.80  
## 309 1769 175402.71  
## 310 1770 163954.57  
## 311 1771 121534.35  
## 312 1772 129379.66  
## 313 1773 120984.97  
## 314 1774 134874.97  
## 315 1775 140418.01  
## 316 1776 135666.97  
## 317 1777 110712.68  
## 318 1778 148592.12  
## 319 1779 104808.65  
## 320 1780 174977.59  
## 321 1781 126910.84  
## 322 1782 84518.45  
## 323 1783 132348.11  
## 324 1784 95981.36  
## 325 1785 129212.48  
## 326 1786 117098.99  
## 327 1787 176220.42  
## 328 1788 62470.56  
## 329 1789 113261.02  
## 330 1790 70690.00  
## 331 1791 190684.57  
## 332 1792 166905.65  
## 333 1793 138843.77  
## 334 1794 171716.23  
## 335 1795 135345.55  
## 336 1796 134865.01  
## 337 1797 132398.60  
## 338 1798 115749.02  
## 339 1799 112476.05  
## 340 1800 124335.31  
## 341 1801 126374.07  
## 342 1802 137103.87  
## 343 1803 149082.09  
## 344 1804 152398.56  
## 345 1805 141234.20  
## 346 1806 130923.53  
## 347 1807 139201.78  
## 348 1808 130945.43  
## 349 1809 119582.22  
## 350 1810 139150.67  
## 351 1811 99334.45  
## 352 1812 93614.08  
## 353 1813 122682.17  
## 354 1814 89951.35  
## 355 1815 57788.86  
## 356 1816 103971.38  
## 357 1817 101983.24  
## 358 1818 147301.36  
## 359 1819 118079.34  
## 360 1820 65744.88  
## 361 1821 106186.48  
## 362 1822 140495.69  
## 363 1823 60571.96  
## 364 1824 132797.25  
## 365 1825 131349.57  
## 366 1826 97374.24  
## 367 1827 101803.64  
## 368 1828 130727.23  
## 369 1829 129551.14  
## 370 1830 141087.41  
## 371 1831 172864.45  
## 372 1832 87035.67  
## 373 1833 145749.27  
## 374 1834 119326.36  
## 375 1835 124494.35  
## 376 1836 128764.15  
## 377 1837 99298.26  
## 378 1838 127282.09  
## 379 1839 96902.24  
## 380 1840 158353.95  
## 381 1841 139036.17  
## 382 1842 93665.36  
## 383 1843 133150.51  
## 384 1844 141717.04  
## 385 1845 136013.04  
## 386 1846 153254.56  
## 387 1847 157222.68  
## 388 1848 54641.30  
## 389 1849 118788.67  
## 390 1850 118469.59  
## 391 1851 157874.28  
## 392 1852 125473.48  
## 393 1853 135008.14  
## 394 1854 182433.71  
## 395 1855 178909.61  
## 396 1856 222007.86  
## 397 1857 136342.01  
## 398 1858 139825.99  
## 399 1859 122272.97  
## 400 1860 148111.40  
## 401 1861 121564.81  
## 402 1862 341187.50  
## 403 1863 336342.36  
## 404 1864 335832.51  
## 405 1865 330880.32  
## 406 1866 346929.70  
## 407 1867 228215.10  
## 408 1868 309727.07  
## 409 1869 205874.11  
## 410 1870 229902.13  
## 411 1871 256915.27  
## 412 1872 170431.85  
## 413 1873 226113.14  
## 414 1874 153551.13  
## 415 1875 208920.04  
## 416 1876 188309.69  
## 417 1877 222420.55  
## 418 1878 214642.34  
## 419 1879 127230.48  
## 420 1880 139465.63  
## 421 1881 242973.84  
## 422 1882 253235.41  
## 423 1883 191590.97  
## 424 1884 203482.51  
## 425 1885 252400.89  
## 426 1886 299700.79  
## 427 1887 223039.68  
## 428 1888 287907.08  
## 429 1889 163384.95  
## 430 1890 123569.09  
## 431 1891 147148.79  
## 432 1892 97803.49  
## 433 1893 132415.42  
## 434 1894 126222.99  
## 435 1895 143163.26  
## 436 1896 126940.54  
## 437 1897 109632.89  
## 438 1898 111123.91  
## 439 1899 150655.75  
## 440 1900 133823.70  
## 441 1901 152085.12  
## 442 1902 133079.75  
## 443 1903 195798.54  
## 444 1904 127187.61  
## 445 1905 181121.97  
## 446 1906 158330.47  
## 447 1907 189626.76  
## 448 1908 107936.58  
## 449 1909 125931.76  
## 450 1910 115840.48  
## 451 1911 212505.90  
## 452 1912 333056.12  
## 453 1913 131420.31  
## 454 1914 80323.50  
## 455 1915 284330.14  
## 456 1916 64503.70  
## 457 1917 257076.87  
## 458 1918 143076.48  
## 459 1919 193799.63  
## 460 1920 166810.68  
## 461 1921 390860.58  
## 462 1922 340464.79  
## 463 1923 246217.82  
## 464 1924 213097.67  
## 465 1925 214738.00  
## 466 1926 383747.94  
## 467 1927 140533.67  
## 468 1928 156199.28  
## 469 1929 130736.46  
## 470 1930 136609.48  
## 471 1931 142244.87  
## 472 1932 131470.25  
## 473 1933 164406.08  
## 474 1934 175782.69  
## 475 1935 156981.10  
## 476 1936 178217.04  
## 477 1937 180032.62  
## 478 1938 182174.26  
## 479 1939 248935.28  
## 480 1940 190945.05  
## 481 1941 173953.38  
## 482 1942 187107.17  
## 483 1943 199257.52  
## 484 1944 343053.68  
## 485 1945 376183.08  
## 486 1946 143933.91  
## 487 1947 318379.59  
## 488 1948 166536.94  
## 489 1949 265171.52  
## 490 1950 204997.67  
## 491 1951 263041.51  
## 492 1952 223192.00  
## 493 1953 174749.16  
## 494 1954 211036.18  
## 495 1955 130920.28  
## 496 1956 377177.54  
## 497 1957 160853.91  
## 498 1958 304475.79  
## 499 1959 139526.93  
## 500 1960 114469.57  
## 501 1961 117047.22  
## 502 1962 93607.60  
## 503 1963 99711.95  
## 504 1964 102777.32  
## 505 1965 140839.87  
## 506 1966 145311.21  
## 507 1967 285102.52  
## 508 1968 385190.58  
## 509 1969 375834.86  
## 510 1970 440151.65  
## 511 1971 524597.62  
## 512 1972 388602.96  
## 513 1973 255356.11  
## 514 1974 343226.75  
## 515 1975 511042.56  
## 516 1976 260666.99  
## 517 1977 374654.19  
## 518 1978 343872.70  
## 519 1979 305757.17  
## 520 1980 196836.32  
## 521 1981 342329.40  
## 522 1982 221475.51  
## 523 1983 202969.06  
## 524 1984 164842.54  
## 525 1985 226612.60  
## 526 1986 224905.62  
## 527 1987 175216.39  
## 528 1988 174609.61  
## 529 1989 194736.02  
## 530 1990 208341.36  
## 531 1991 223447.86  
## 532 1992 210381.17  
## 533 1993 173937.08  
## 534 1994 251718.68  
## 535 1995 179659.38  
## 536 1996 244504.14  
## 537 1997 325558.60  
## 538 1998 290799.30  
## 539 1999 275072.61  
## 540 2000 312526.01  
## 541 2001 245394.05  
## 542 2002 277720.04  
## 543 2003 256424.69  
## 544 2004 257738.53  
## 545 2005 250783.63  
## 546 2006 204973.05  
## 547 2007 240821.57  
## 548 2008 198635.57  
## 549 2009 186450.68  
## 550 2010 196508.78  
## 551 2011 153950.61  
## 552 2012 179939.88  
## 553 2013 171570.17  
## 554 2014 183588.25  
## 555 2015 203456.08  
## 556 2016 186785.80  
## 557 2017 190968.82  
## 558 2018 117294.39  
## 559 2019 139704.22  
## 560 2020 118679.65  
## 561 2021 107650.14  
## 562 2022 183555.67  
## 563 2023 146387.23  
## 564 2024 279437.71  
## 565 2025 323384.60  
## 566 2026 169527.24  
## 567 2027 144085.70  
## 568 2028 159536.85  
## 569 2029 149551.60  
## 570 2030 260483.81  
## 571 2031 244135.71  
## 572 2032 260536.73  
## 573 2033 253167.28  
## 574 2034 160597.44  
## 575 2035 204038.74  
## 576 2036 195811.25  
## 577 2037 199880.33  
## 578 2038 316456.41  
## 579 2039 210860.08  
## 580 2040 315722.75  
## 581 2041 325945.91  
## 582 2042 195312.25  
## 583 2043 184231.66  
## 584 2044 178596.77  
## 585 2045 217930.09  
## 586 2046 145508.07  
## 587 2047 139467.63  
## 588 2048 136363.75  
## 589 2049 143296.17  
## 590 2050 180485.45  
## 591 2051 102758.05  
## 592 2052 131277.58  
## 593 2053 145396.10  
## 594 2054 84016.56  
## 595 2055 159226.95  
## 596 2056 153593.85  
## 597 2057 113204.97  
## 598 2058 219869.54  
## 599 2059 128490.82  
## 600 2060 181214.37  
## 601 2061 181268.40  
## 602 2062 125424.78  
## 603 2063 119465.71  
## 604 2064 131051.54  
## 605 2065 121208.83  
## 606 2066 173388.63  
## 607 2067 115947.21  
## 608 2068 144246.93  
## 609 2069 90524.21  
## 610 2070 113823.83  
## 611 2071 91540.55  
## 612 2072 136745.18  
## 613 2073 140285.14  
## 614 2074 179734.04  
## 615 2075 143175.69  
## 616 2076 118622.94  
## 617 2077 137618.01  
## 618 2078 115679.12  
## 619 2079 137804.52  
## 620 2080 127602.77  
## 621 2081 125089.21  
## 622 2082 125007.79  
## 623 2083 140854.77  
## 624 2084 110323.23  
## 625 2085 115058.53  
## 626 2086 150798.14  
## 627 2087 134416.99  
## 628 2088 89893.70  
## 629 2089 76214.60  
## 630 2090 123468.40  
## 631 2091 95261.05  
## 632 2092 129626.01  
## 633 2093 126822.06  
## 634 2094 118501.29  
## 635 2095 144153.45  
## 636 2096 94027.78  
## 637 2097 94897.45  
## 638 2098 144589.74  
## 639 2099 60211.38  
## 640 2100 86446.08  
## 641 2101 99346.53  
## 642 2102 132668.29  
## 643 2103 96603.27  
## 644 2104 144667.15  
## 645 2105 119784.08  
## 646 2106 63360.86  
## 647 2107 188541.17  
## 648 2108 123307.88  
## 649 2109 112839.05  
## 650 2110 126382.78  
## 651 2111 164283.80  
## 652 2112 130795.20  
## 653 2113 121696.10  
## 654 2114 118543.89  
## 655 2115 155027.40  
## 656 2116 112858.53  
## 657 2117 134922.33  
## 658 2118 114298.58  
## 659 2119 113634.93  
## 660 2120 102471.17  
## 661 2121 89727.88  
## 662 2122 96134.14  
## 663 2123 88961.35  
## 664 2124 170445.92  
## 665 2125 133849.27  
## 666 2126 132761.64  
## 667 2127 160351.91  
## 668 2128 129154.74  
## 669 2129 108517.18  
## 670 2130 137741.53  
## 671 2131 139021.83  
## 672 2132 117809.51  
## 673 2133 124119.20  
## 674 2134 121084.40  
## 675 2135 102588.31  
## 676 2136 72841.20  
## 677 2137 118434.26  
## 678 2138 149899.38  
## 679 2139 153915.92  
## 680 2140 142637.53  
## 681 2141 164842.20  
## 682 2142 119522.90  
## 683 2143 134557.52  
## 684 2144 121317.50  
## 685 2145 142285.15  
## 686 2146 178371.70  
## 687 2147 156726.56  
## 688 2148 146613.27  
## 689 2149 158102.53  
## 690 2150 220926.80  
## 691 2151 110228.17  
## 692 2152 164602.15  
## 693 2153 187907.89  
## 694 2154 116786.42  
## 695 2155 144393.04  
## 696 2156 276256.91  
## 697 2157 230025.13  
## 698 2158 240897.92  
## 699 2159 223635.20  
## 700 2160 175059.27  
## 701 2161 237138.44  
## 702 2162 409028.61  
## 703 2163 359797.91  
## 704 2164 245857.31  
## 705 2165 183501.46  
## 706 2166 158028.04  
## 707 2167 211929.51  
## 708 2168 197861.37  
## 709 2169 192933.38  
## 710 2170 220499.58  
## 711 2171 155233.49  
## 712 2172 139051.81  
## 713 2173 183826.07  
## 714 2174 233420.45  
## 715 2175 298545.02  
## 716 2176 325386.30  
## 717 2177 245683.87  
## 718 2178 215746.07  
## 719 2179 146409.90  
## 720 2180 214824.54  
## 721 2181 196000.77  
## 722 2182 228413.16  
## 723 2183 193054.76  
## 724 2184 129542.39  
## 725 2185 133802.71  
## 726 2186 155767.86  
## 727 2187 155686.21  
## 728 2188 151582.71  
## 729 2189 268109.56  
## 730 2190 84792.03  
## 731 2191 85568.66  
## 732 2192 94379.05  
## 733 2193 110534.76  
## 734 2194 107594.29  
## 735 2195 106543.64  
## 736 2196 127058.79  
## 737 2197 108867.91  
## 738 2198 156911.05  
## 739 2199 165555.89  
## 740 2200 128305.30  
## 741 2201 119904.06  
## 742 2202 181611.64  
## 743 2203 115415.51  
## 744 2204 161799.16  
## 745 2205 110860.07  
## 746 2206 133478.16  
## 747 2207 180338.02  
## 748 2208 261226.42  
## 749 2209 194447.20  
## 750 2210 113014.33  
## 751 2211 124841.73  
## 752 2212 116516.22  
## 753 2213 107661.86  
## 754 2214 137878.19  
## 755 2215 106896.35  
## 756 2216 148050.49  
## 757 2217 72032.44  
## 758 2218 84548.32  
## 759 2219 78742.02  
## 760 2220 74728.21  
## 761 2221 284330.14  
## 762 2222 269548.67  
## 763 2223 305533.03  
## 764 2224 205323.38  
## 765 2225 121642.62  
## 766 2226 180104.55  
## 767 2227 207268.13  
## 768 2228 281613.06  
## 769 2229 244034.51  
## 770 2230 145622.17  
## 771 2231 202840.98  
## 772 2232 175204.11  
## 773 2233 190553.71  
## 774 2234 263507.84  
## 775 2235 233288.77  
## 776 2236 260390.79  
## 777 2237 328391.19  
## 778 2238 205027.25  
## 779 2239 127458.45  
## 780 2240 177522.77  
## 781 2241 146705.30  
## 782 2242 130322.84  
## 783 2243 139875.57  
## 784 2244 98204.92  
## 785 2245 93021.56  
## 786 2246 151710.75  
## 787 2247 116451.56  
## 788 2248 123913.16  
## 789 2249 123720.22  
## 790 2250 136799.30  
## 791 2251 108622.26  
## 792 2252 185052.21  
## 793 2253 154508.49  
## 794 2254 175741.80  
## 795 2255 190290.37  
## 796 2256 177673.39  
## 797 2257 203583.20  
## 798 2258 169734.50  
## 799 2259 165392.98  
## 800 2260 157655.06  
## 801 2261 183052.34  
## 802 2262 202187.68  
## 803 2263 379076.71  
## 804 2264 492038.47  
## 805 2265 179112.67  
## 806 2266 297305.75  
## 807 2267 326812.69  
## 808 2268 424243.73  
## 809 2269 162799.28  
## 810 2270 197592.74  
## 811 2271 230488.08  
## 812 2272 217019.29  
## 813 2273 175622.59  
## 814 2274 175435.64  
## 815 2275 154518.71  
## 816 2276 190099.46  
## 817 2277 194331.09  
## 818 2278 156719.13  
## 819 2279 118436.81  
## 820 2280 120128.74  
## 821 2281 168466.99  
## 822 2282 202041.05  
## 823 2283 105036.10  
## 824 2284 114040.33  
## 825 2285 147846.40  
## 826 2286 125820.44  
## 827 2287 354149.52  
## 828 2288 268935.81  
## 829 2289 311417.85  
## 830 2290 437794.34  
## 831 2291 335781.34  
## 832 2292 443739.35  
## 833 2293 440493.43  
## 834 2294 384532.54  
## 835 2295 464367.58  
## 836 2296 263335.52  
## 837 2297 329663.84  
## 838 2298 318348.85  
## 839 2299 341791.78  
## 840 2300 342160.50  
## 841 2301 295438.77  
## 842 2302 242381.22  
## 843 2303 231163.34  
## 844 2304 240382.01  
## 845 2305 178645.40  
## 846 2306 180953.41  
## 847 2307 193830.83  
## 848 2308 209779.47  
## 849 2309 302552.81  
## 850 2310 201898.18  
## 851 2311 198482.15  
## 852 2312 185430.35  
## 853 2313 177170.74  
## 854 2314 171848.56  
## 855 2315 172249.01  
## 856 2316 195634.73  
## 857 2317 192415.65  
## 858 2318 167834.99  
## 859 2319 175551.14  
## 860 2320 193477.21  
## 861 2321 237916.55  
## 862 2322 176294.87  
## 863 2323 203101.87  
## 864 2324 187352.16  
## 865 2325 209593.15  
## 866 2326 183809.27  
## 867 2327 206129.24  
## 868 2328 219296.36  
## 869 2329 187397.33  
## 870 2330 174296.50  
## 871 2331 332777.94  
## 872 2332 413361.71  
## 873 2333 323220.40  
## 874 2334 252547.91  
## 875 2335 293976.59  
## 876 2336 306877.59  
## 877 2337 204121.84  
## 878 2338 265517.69  
## 879 2339 222836.60  
## 880 2340 403880.76  
## 881 2341 212291.36  
## 882 2342 231409.59  
## 883 2343 219005.20  
## 884 2344 224229.88  
## 885 2345 237472.76  
## 886 2346 202967.94  
## 887 2347 189336.31  
## 888 2348 219219.61  
## 889 2349 166586.02  
## 890 2350 301163.87  
## 891 2351 239989.71  
## 892 2352 242641.60  
## 893 2353 244732.76  
## 894 2354 134124.48  
## 895 2355 154925.02  
## 896 2356 166106.38  
## 897 2357 194567.18  
## 898 2358 192924.81  
## 899 2359 127621.71  
## 900 2360 113078.56  
## 901 2361 143557.65  
## 902 2362 297790.99  
## 903 2363 131475.06  
## 904 2364 181173.03  
## 905 2365 215202.56  
## 906 2366 178820.67  
## 907 2367 206985.12  
## 908 2368 219417.27  
## 909 2369 207524.85  
## 910 2370 151685.93  
## 911 2371 154718.68  
## 912 2372 183016.40  
## 913 2373 307931.14  
## 914 2374 323670.02  
## 915 2375 233156.76  
## 916 2376 283931.42  
## 917 2377 310021.32  
## 918 2378 148875.31  
## 919 2379 223024.41  
## 920 2380 154109.32  
## 921 2381 166641.42  
## 922 2382 224739.74  
## 923 2383 204991.27  
## 924 2384 252676.04  
## 925 2385 169221.38  
## 926 2386 132932.26  
## 927 2387 134819.95  
## 928 2388 110403.80  
## 929 2389 126967.95  
## 930 2390 141141.67  
## 931 2391 137085.82  
## 932 2392 117613.96  
## 933 2393 169242.31  
## 934 2394 137200.06  
## 935 2395 202011.07  
## 936 2396 141815.79  
## 937 2397 232443.26  
## 938 2398 128533.53  
## 939 2399 64322.52  
## 940 2400 65218.64  
## 941 2401 114842.20  
## 942 2402 135339.64  
## 943 2403 143039.08  
## 944 2404 153615.71  
## 945 2405 156856.00  
## 946 2406 143344.04  
## 947 2407 125932.78  
## 948 2408 148171.54  
## 949 2409 119958.17  
## 950 2410 182989.94  
## 951 2411 114274.96  
## 952 2412 147841.83  
## 953 2413 124591.21  
## 954 2414 162319.49  
## 955 2415 130802.39  
## 956 2416 132945.47  
## 957 2417 130379.73  
## 958 2418 132006.49  
## 959 2419 114229.27  
## 960 2420 120314.31  
## 961 2421 151584.92  
## 962 2422 113226.67  
## 963 2423 106787.67  
## 964 2424 135472.02  
## 965 2425 221296.66  
## 966 2426 130970.54  
## 967 2427 122197.40  
## 968 2428 173341.02  
## 969 2429 119229.15  
## 970 2430 137662.45  
## 971 2431 109589.10  
## 972 2432 134706.92  
## 973 2433 139361.40  
## 974 2434 129304.71  
## 975 2435 144925.31  
## 976 2436 99083.21  
## 977 2437 103581.44  
## 978 2438 117636.48  
## 979 2439 116099.07  
## 980 2440 122812.54  
## 981 2441 100709.54  
## 982 2442 95019.65  
## 983 2443 136124.86  
## 984 2444 114856.36  
## 985 2445 83225.89  
## 986 2446 136112.88  
## 987 2447 198029.69  
## 988 2448 136592.10  
## 989 2449 102090.91  
## 990 2450 154080.41  
## 991 2451 112986.68  
## 992 2452 193436.83  
## 993 2453 95550.54  
## 994 2454 122003.21  
## 995 2455 121891.93  
## 996 2456 129603.03  
## 997 2457 118629.60  
## 998 2458 115943.91  
## 999 2459 110313.72  
## 1000 2460 127796.45  
## 1001 2461 128824.60  
## 1002 2462 123984.68  
## 1003 2463 122407.15  
## 1004 2464 170223.54  
## 1005 2465 131081.24  
## 1006 2466 106868.64  
## 1007 2467 136842.49  
## 1008 2468 90023.88  
## 1009 2469 84360.59  
## 1010 2470 212329.99  
## 1011 2471 226890.82  
## 1012 2472 156290.95  
## 1013 2473 118163.15  
## 1014 2474 81329.61  
## 1015 2475 225745.64  
## 1016 2476 112580.36  
## 1017 2477 126779.21  
## 1018 2478 156729.44  
## 1019 2479 111286.96  
## 1020 2480 159467.29  
## 1021 2481 116200.25  
## 1022 2482 130833.28  
## 1023 2483 109312.43  
## 1024 2484 119226.60  
## 1025 2485 128891.90  
## 1026 2486 139653.73  
## 1027 2487 176418.06  
## 1028 2488 160932.13  
## 1029 2489 153961.07  
## 1030 2490 139954.45  
## 1031 2491 102408.93  
## 1032 2492 185133.06  
## 1033 2493 163728.23  
## 1034 2494 164023.04  
## 1035 2495 115132.79  
## 1036 2496 243684.49  
## 1037 2497 154323.94  
## 1038 2498 103048.20  
## 1039 2499 95885.98  
## 1040 2500 129181.66  
## 1041 2501 130986.72  
## 1042 2502 150468.02  
## 1043 2503 100432.68  
## 1044 2504 207637.26  
## 1045 2505 223826.08  
## 1046 2506 238210.75  
## 1047 2507 257681.24  
## 1048 2508 241233.59  
## 1049 2509 215872.44  
## 1050 2510 217184.06  
## 1051 2511 182006.20  
## 1052 2512 198821.02  
## 1053 2513 223361.06  
## 1054 2514 236184.14  
## 1055 2515 142754.91  
## 1056 2516 168001.44  
## 1057 2517 145143.88  
## 1058 2518 146997.80  
## 1059 2519 219276.21  
## 1060 2520 213261.67  
## 1061 2521 191420.52  
## 1062 2522 226052.07  
## 1063 2523 125764.45  
## 1064 2524 136824.35  
## 1065 2525 154082.21  
## 1066 2526 132559.09  
## 1067 2527 118835.95  
## 1068 2528 126141.94  
## 1069 2529 130311.92  
## 1070 2530 131408.83  
## 1071 2531 249497.26  
## 1072 2532 229211.94  
## 1073 2533 191692.59  
## 1074 2534 227361.47  
## 1075 2535 308790.07  
## 1076 2536 257197.57  
## 1077 2537 231619.00  
## 1078 2538 177427.23  
## 1079 2539 195465.08  
## 1080 2540 180649.53  
## 1081 2541 182689.56  
## 1082 2542 159926.19  
## 1083 2543 126635.11  
## 1084 2544 120392.14  
## 1085 2545 135895.95  
## 1086 2546 138349.15  
## 1087 2547 147959.25  
## 1088 2548 151905.44  
## 1089 2549 151444.23  
## 1090 2550 806080.29  
## 1091 2551 150905.46  
## 1092 2552 122438.87  
## 1093 2553 78080.23  
## 1094 2554 110134.02  
## 1095 2555 120988.96  
## 1096 2556 98032.80  
## 1097 2557 108888.91  
## 1098 2558 157363.65  
## 1099 2559 115927.36  
## 1100 2560 124127.46  
## 1101 2561 122762.71  
## 1102 2562 126167.53  
## 1103 2563 130549.14  
## 1104 2564 172537.97  
## 1105 2565 121939.08  
## 1106 2566 131140.21  
## 1107 2567 122102.68  
## 1108 2568 172979.86  
## 1109 2569 199981.26  
## 1110 2570 113214.24  
## 1111 2571 171074.39  
## 1112 2572 157827.50  
## 1113 2573 206126.51  
## 1114 2574 286087.35  
## 1115 2575 134313.91  
## 1116 2576 125761.69  
## 1117 2577 148695.54  
## 1118 2578 93792.37  
## 1119 2579 61974.95  
## 1120 2580 109679.60  
## 1121 2581 125570.27  
## 1122 2582 136251.32  
## 1123 2583 260390.07  
## 1124 2584 193993.81  
## 1125 2585 189050.34  
## 1126 2586 237586.05  
## 1127 2587 205741.51  
## 1128 2588 147083.05  
## 1129 2589 155746.92  
## 1130 2590 213456.18  
## 1131 2591 237796.14  
## 1132 2592 206771.05  
## 1133 2593 239722.05  
## 1134 2594 194063.79  
## 1135 2595 216001.29  
## 1136 2596 287468.30  
## 1137 2597 193561.57  
## 1138 2598 281271.45  
## 1139 2599 351684.92  
## 1140 2600 156285.92  
## 1141 2601 149913.56  
## 1142 2602 79933.37  
## 1143 2603 85600.77  
## 1144 2604 84868.82  
## 1145 2605 76951.42  
## 1146 2606 144687.58  
## 1147 2607 198988.77  
## 1148 2608 218842.68  
## 1149 2609 166415.23  
## 1150 2610 116841.28  
## 1151 2611 131351.75  
## 1152 2612 163692.20  
## 1153 2613 135863.57  
## 1154 2614 127270.24  
## 1155 2615 148439.89  
## 1156 2616 161059.51  
## 1157 2617 174907.69  
## 1158 2618 207388.30  
## 1159 2619 196993.45  
## 1160 2620 172151.80  
## 1161 2621 170283.97  
## 1162 2622 173581.51  
## 1163 2623 250479.84  
## 1164 2624 282633.90  
## 1165 2625 278058.69  
## 1166 2626 179429.69  
## 1167 2627 160795.96  
## 1168 2628 464544.06  
## 1169 2629 527500.78  
## 1170 2630 344671.90  
## 1171 2631 480712.27  
## 1172 2632 406502.38  
## 1173 2633 300416.92  
## 1174 2634 413836.71  
## 1175 2635 165557.93  
## 1176 2636 181303.30  
## 1177 2637 188873.82  
## 1178 2638 255162.85  
## 1179 2639 202024.23  
## 1180 2640 156394.69  
## 1181 2641 99860.59  
## 1182 2642 200945.66  
## 1183 2643 104289.96  
## 1184 2644 127941.48  
## 1185 2645 105169.93  
## 1186 2646 91174.44  
## 1187 2647 97259.57  
## 1188 2648 150118.56  
## 1189 2649 152181.45  
## 1190 2650 142986.95  
## 1191 2651 134254.12  
## 1192 2652 437560.88  
## 1193 2653 243726.93  
## 1194 2654 247330.16  
## 1195 2655 387197.77  
## 1196 2656 298493.55  
## 1197 2657 356622.13  
## 1198 2658 298802.60  
## 1199 2659 302798.76  
## 1200 2660 358282.79  
## 1201 2661 338554.18  
## 1202 2662 356727.28  
## 1203 2663 283987.97  
## 1204 2664 284965.96  
## 1205 2665 322326.20  
## 1206 2666 281432.00  
## 1207 2667 171851.32  
## 1208 2668 181892.93  
## 1209 2669 179286.54  
## 1210 2670 272731.01  
## 1211 2671 179055.41  
## 1212 2672 189199.15  
## 1213 2673 203547.69  
## 1214 2674 206881.88  
## 1215 2675 161783.06  
## 1216 2676 193781.85  
## 1217 2677 194386.46  
## 1218 2678 258322.22  
## 1219 2679 263192.70  
## 1220 2680 278652.04  
## 1221 2681 425019.64  
## 1222 2682 306014.18  
## 1223 2683 515633.81  
## 1224 2684 292085.67  
## 1225 2685 319745.98  
## 1226 2686 242292.09  
## 1227 2687 322181.80  
## 1228 2688 215231.43  
## 1229 2689 198764.86  
## 1230 2690 453909.93  
## 1231 2691 180307.26  
## 1232 2692 138784.70  
## 1233 2693 194919.61  
## 1234 2694 147270.91  
## 1235 2695 191941.94  
## 1236 2696 183966.23  
## 1237 2697 172930.78  
## 1238 2698 203613.04  
## 1239 2699 186362.33  
## 1240 2700 164753.77  
## 1241 2701 149092.56  
## 1242 2702 113757.57  
## 1243 2703 142254.12  
## 1244 2704 143564.91  
## 1245 2705 116834.65  
## 1246 2706 116460.99  
## 1247 2707 124166.36  
## 1248 2708 131007.66  
## 1249 2709 112600.84  
## 1250 2710 125805.33  
## 1251 2711 313605.46  
## 1252 2712 387575.97  
## 1253 2713 164034.21  
## 1254 2714 136093.48  
## 1255 2715 156362.55  
## 1256 2716 140428.37  
## 1257 2717 185205.86  
## 1258 2718 213973.03  
## 1259 2719 161766.63  
## 1260 2720 192506.12  
## 1261 2721 132550.61  
## 1262 2722 160378.54  
## 1263 2723 161477.10  
## 1264 2724 124988.38  
## 1265 2725 126015.68  
## 1266 2726 142629.53  
## 1267 2727 193889.69  
## 1268 2728 167513.43  
## 1269 2729 135987.57  
## 1270 2730 146616.86  
## 1271 2731 129830.18  
## 1272 2732 124481.85  
## 1273 2733 160156.99  
## 1274 2734 146758.20  
## 1275 2735 129202.00  
## 1276 2736 137994.58  
## 1277 2737 116765.29  
## 1278 2738 135920.47  
## 1279 2739 163653.85  
## 1280 2740 139081.53  
## 1281 2741 126320.79  
## 1282 2742 151349.40  
## 1283 2743 148063.51  
## 1284 2744 153537.51  
## 1285 2745 154981.95  
## 1286 2746 127881.43  
## 1287 2747 163022.93  
## 1288 2748 137850.41  
## 1289 2749 130183.67  
## 1290 2750 126314.82  
## 1291 2751 130594.22  
## 1292 2752 211383.81  
## 1293 2753 144505.42  
## 1294 2754 216476.49  
## 1295 2755 132011.66  
## 1296 2756 95586.47  
## 1297 2757 77482.10  
## 1298 2758 74505.12  
## 1299 2759 148088.08  
## 1300 2760 105392.14  
## 1301 2761 136263.65  
## 1302 2762 136016.67  
## 1303 2763 207038.57  
## 1304 2764 163454.42  
## 1305 2765 291138.03  
## 1306 2766 130386.81  
## 1307 2767 81478.22  
## 1308 2768 119513.21  
## 1309 2769 121366.94  
## 1310 2770 153416.71  
## 1311 2771 112488.92  
## 1312 2772 115427.66  
## 1313 2773 147437.55  
## 1314 2774 143037.47  
## 1315 2775 116935.16  
## 1316 2776 142319.60  
## 1317 2777 133744.64  
## 1318 2778 113657.19  
## 1319 2779 120084.12  
## 1320 2780 107785.69  
## 1321 2781 99848.18  
## 1322 2782 96676.01  
## 1323 2783 89315.24  
## 1324 2784 130218.22  
## 1325 2785 127599.31  
## 1326 2786 65349.39  
## 1327 2787 126957.46  
## 1328 2788 74539.67  
## 1329 2789 159053.40  
## 1330 2790 93353.37  
## 1331 2791 115146.41  
## 1332 2792 69609.61  
## 1333 2793 150951.75  
## 1334 2794 91147.84  
## 1335 2795 114755.25  
## 1336 2796 95209.43  
## 1337 2797 219955.89  
## 1338 2798 118073.32  
## 1339 2799 103071.81  
## 1340 2800 85049.42  
## 1341 2801 114369.82  
## 1342 2802 133081.51  
## 1343 2803 156288.82  
## 1344 2804 134891.78  
## 1345 2805 105473.73  
## 1346 2806 84327.01  
## 1347 2807 158376.40  
## 1348 2808 143202.20  
## 1349 2809 135955.91  
## 1350 2810 132184.80  
## 1351 2811 147054.83  
## 1352 2812 162522.77  
## 1353 2813 146626.76  
## 1354 2814 148935.09  
## 1355 2815 103800.51  
## 1356 2816 225513.93  
## 1357 2817 139677.39  
## 1358 2818 135211.74  
## 1359 2819 161154.77  
## 1360 2820 131583.76  
## 1361 2821 108380.99  
## 1362 2822 209123.14  
## 1363 2823 338109.09  
## 1364 2824 157781.77  
## 1365 2825 145602.83  
## 1366 2826 137089.35  
## 1367 2827 140634.92  
## 1368 2828 254668.13  
## 1369 2829 226208.98  
## 1370 2830 223809.29  
## 1371 2831 184626.51  
## 1372 2832 246379.45  
## 1373 2833 298782.89  
## 1374 2834 204280.47  
## 1375 2835 209654.63  
## 1376 2836 185573.76  
## 1377 2837 173061.93  
## 1378 2838 147566.30  
## 1379 2839 174896.74  
## 1380 2840 194309.69  
## 1381 2841 200496.06  
## 1382 2842 230328.23  
## 1383 2843 140624.70  
## 1384 2844 177497.16  
## 1385 2845 129442.28  
## 1386 2846 204702.35  
## 1387 2847 188441.90  
## 1388 2848 225684.91  
## 1389 2849 201338.43  
## 1390 2850 286645.29  
## 1391 2851 236621.81  
## 1392 2852 233948.51  
## 1393 2853 229687.46  
## 1394 2854 146489.30  
## 1395 2855 210840.46  
## 1396 2856 194287.28  
## 1397 2857 184934.36  
## 1398 2858 209144.67  
## 1399 2859 111642.12  
## 1400 2860 142674.97  
## 1401 2861 139126.31  
## 1402 2862 192006.79  
## 1403 2863 135370.76  
## 1404 2864 266220.96  
## 1405 2865 147148.79  
## 1406 2866 177758.27  
## 1407 2867 92870.71  
## 1408 2868 99089.27  
## 1409 2869 110106.82  
## 1410 2870 139971.06  
## 1411 2871 84396.43  
## 1412 2872 55074.97  
## 1413 2873 112108.38  
## 1414 2874 132974.90  
## 1415 2875 131771.23  
## 1416 2876 159830.97  
## 1417 2877 126801.02  
## 1418 2878 156559.41  
## 1419 2879 132364.95  
## 1420 2880 97787.11  
## 1421 2881 132083.47  
## 1422 2882 152414.78  
## 1423 2883 178777.00  
## 1424 2884 188136.28  
## 1425 2885 165457.51  
## 1426 2886 226404.75  
## 1427 2887 113500.03  
## 1428 2888 131552.62  
## 1429 2889 68562.28  
## 1430 2890 82413.56  
## 1431 2891 143773.65  
## 1432 2892 71871.19  
## 1433 2893 114424.69  
## 1434 2894 67362.81  
## 1435 2895 319938.59  
## 1436 2896 278145.17  
## 1437 2897 211946.73  
## 1438 2898 153236.10  
## 1439 2899 230128.35  
## 1440 2900 161948.85  
## 1441 2901 205693.59  
## 1442 2902 196693.86  
## 1443 2903 334713.49  
## 1444 2904 328107.39  
## 1445 2905 88243.84  
## 1446 2906 202112.07  
## 1447 2907 111157.77  
## 1448 2908 129119.83  
## 1449 2909 153076.21  
## 1450 2910 82425.41  
## 1451 2911 79828.99  
## 1452 2912 140940.82  
## 1453 2913 84935.75  
## 1454 2914 78287.46  
## 1455 2915 89048.40  
## 1456 2916 85650.04  
## 1457 2917 168733.63  
## 1458 2918 121410.90  
## 1459 2919 221366.41

write.table(prediccion, file="SalePrice Prediction", sep="\t", col.names=TRUE, row.names=FALSE)

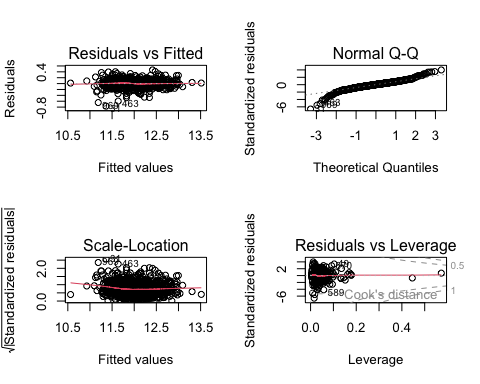
## Interpret the Model

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   
## -0.76751 -0.05963 0.00427 0.06435 0.46504   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.696e+00 5.061e-01 3.350 0.000838 \*\*\*  
## LotFrontage -3.942e-04 2.151e-04 -1.832 0.067238 .   
## log(LotArea) 9.760e-02 1.027e-02 9.507 < 2e-16 \*\*\*  
## OverallQual 6.652e-02 5.199e-03 12.795 < 2e-16 \*\*\*  
## OverallCond 5.278e-02 3.830e-03 13.782 < 2e-16 \*\*\*  
## YearBuilt 3.054e-03 2.240e-04 13.632 < 2e-16 \*\*\*  
## BsmtFinSF1 7.797e-05 1.243e-05 6.274 5.27e-10 \*\*\*  
## log(X1stFlrSF) 3.918e-01 2.409e-02 16.261 < 2e-16 \*\*\*  
## X2ndFlrSF 2.332e-04 1.740e-05 13.400 < 2e-16 \*\*\*  
## BsmtFullBath 2.317e-02 9.672e-03 2.395 0.016789 \*   
## FullBath 2.829e-02 1.059e-02 2.672 0.007653 \*\*   
## TotRmsAbvGrd -1.020e-04 4.577e-03 -0.022 0.982231   
## KitchenAbvGr -7.037e-02 2.014e-02 -3.494 0.000497 \*\*\*  
## Fireplaces 2.419e-02 6.931e-03 3.491 0.000503 \*\*\*  
## GarageCars 4.925e-02 6.932e-03 7.105 2.30e-12 \*\*\*  
## ScreenPorch 2.979e-04 7.342e-05 4.058 5.35e-05 \*\*\*  
## PoolArea 1.474e-04 1.204e-04 1.224 0.221163   
## ExterQualFa -6.558e-02 5.283e-02 -1.241 0.214793   
## ExterQualGd -2.243e-02 2.542e-02 -0.883 0.377666   
## ExterQualTA -5.351e-02 2.828e-02 -1.892 0.058745 .   
## BsmtQualFa -1.558e-01 3.372e-02 -4.621 4.32e-06 \*\*\*  
## BsmtQualGd -9.745e-02 1.901e-02 -5.126 3.55e-07 \*\*\*  
## BsmtQualNo Basement -2.209e-01 3.457e-02 -6.388 2.58e-10 \*\*\*  
## BsmtQualTA -1.070e-01 2.219e-02 -4.821 1.65e-06 \*\*\*  
## KitchenQualFa -6.092e-02 3.513e-02 -1.734 0.083230 .   
## KitchenQualGd -4.293e-02 1.876e-02 -2.289 0.022285 \*   
## KitchenQualTA -7.438e-02 2.100e-02 -3.543 0.000415 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1173 on 989 degrees of freedom  
## Multiple R-squared: 0.9152, Adjusted R-squared: 0.913   
## F-statistic: 410.6 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   
## 1.695567e+00 -3.941671e-04 9.760302e-02 6.652039e-02   
## OverallCond YearBuilt BsmtFinSF1 log(X1stFlrSF)   
## 5.278278e-02 3.054057e-03 7.796773e-05 3.917527e-01   
## X2ndFlrSF BsmtFullBath FullBath TotRmsAbvGrd   
## 2.331492e-04 2.316898e-02 2.829100e-02 -1.019637e-04   
## KitchenAbvGr Fireplaces GarageCars ScreenPorch   
## -7.036863e-02 2.419235e-02 4.925005e-02 2.979136e-04   
## PoolArea ExterQualFa ExterQualGd ExterQualTA   
## 1.473881e-04 -6.558058e-02 -2.243330e-02 -5.351491e-02   
## BsmtQualFa BsmtQualGd BsmtQualNo Basement BsmtQualTA   
## -1.558118e-01 -9.744856e-02 -2.208620e-01 -1.069885e-01   
## KitchenQualFa KitchenQualGd KitchenQualTA   
## -6.092359e-02 -4.293402e-02 -7.438076e-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. Larger first-floor areas are often associated with more spacious living arrangements, which can contribute to higher property values. Continuing with the analysis of the coefficients, the variable “BsmtQualNo Basement” is noteworthy due to its substantial negative impact of -0.227. This suggests that houses without a basement tend to have a significant decrease in their sale price, all else being equal. Basements often contribute to the overall living space and functionality of a house, so the absence of this feature appears to have a notable effect on the property’s value. The presence of a fireplace, as indicated by the variable “Fireplaces,” has a positive coefficient of 0.0248. This suggests that houses with fireplaces tend to have higher predicted sale prices. Fireplaces are often considered desirable features, providing both functional and aesthetic benefits, which can positively impact a property’s perceived value. Variables related to kitchen quality, such as “KitchenQualFa” (Kitchen Quality Fair) and “KitchenQualTA” (Kitchen Quality Typical), both have negative coefficients of -0.064 and -0.075, respectively. This indicates that, compared to higher-quality kitchens, houses with fair or typical kitchen quality are associated with lower predicted sale prices. Kitchens are known to be focal points for homebuyers, and a well-appointed kitchen is often considered a positive attribute. In conclusion, the coefficients shed light on various aspects that influence the predicted sale prices of houses in the model. Factors such as the house square feet, the presence of a fireplace, and the quality of the kitchen and exterior all play roles in determining the estimated market value of a property.

## Annex