Assignment 1

Alicia Chimeno & Silvia Ferrer

2023-11-19

# Explore and Preproces the dataset

We start by looking at an overview of our dataset. Annex.1

dim(train)  
str(train)

## Preprocess and select variables

### Record incorrect NA’s

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

## Recode variables to factor

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

## Target variable exploration “SalePrice”

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

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

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

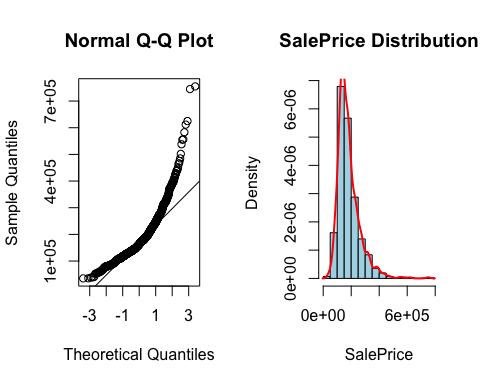
shapiro.test(train$SalePrice)

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

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.

## 

## 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)

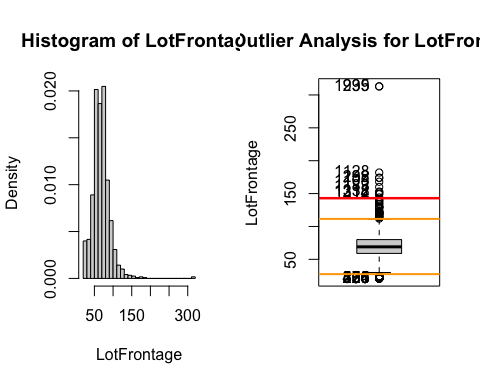
numeric\_description(train2$LotFrontage, 30) # not normal distribution

## 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)

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

## Count of missing values: 0

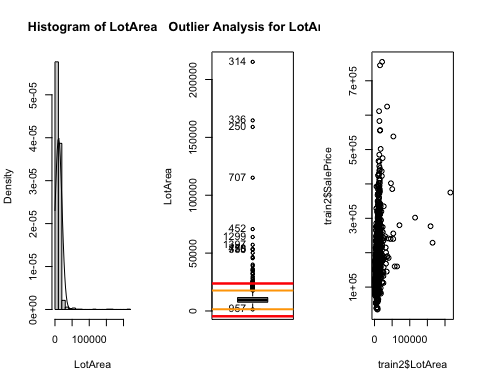
##   
## Shapiro-Wilk normality test  
##   
## data: variable  
## W = 0.35106, p-value < 2.2e-16

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)

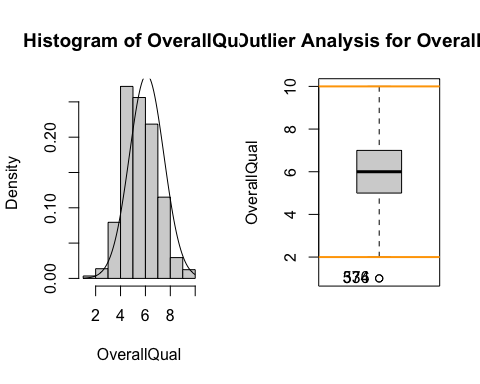
par(mfrow=c(1,2))  
numeric\_description(train2$OverallQual, 10) # not normal distribution

## 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)

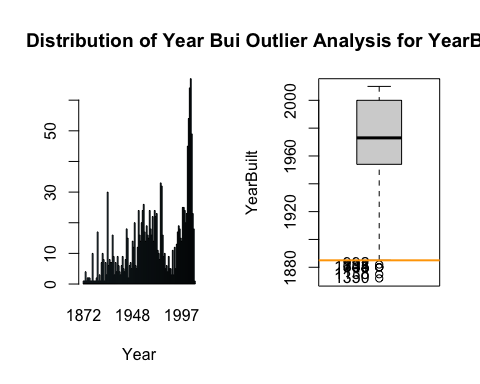
## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1872 1954 1973 1971 2000 2010

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

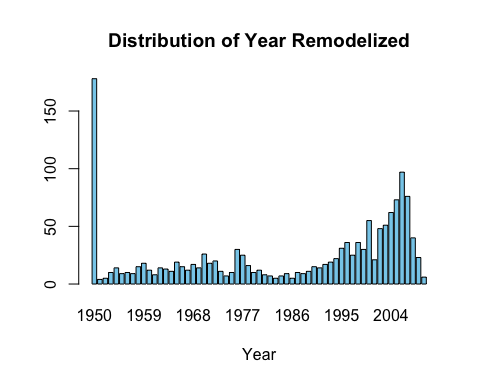
train2 <- update\_outliers\_count(train2, "YearBuilt")

### 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")

Gráfico, Gráfico de cajas y bigotes

Descripción generada automáticamente

sum(is.na(train2$YearRemodAdd))

## [1] 0

analyze\_outliers(train2, "YearRemodAdd")

## Number of mild outliers: 0

## Number of severe outliers: 0

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

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

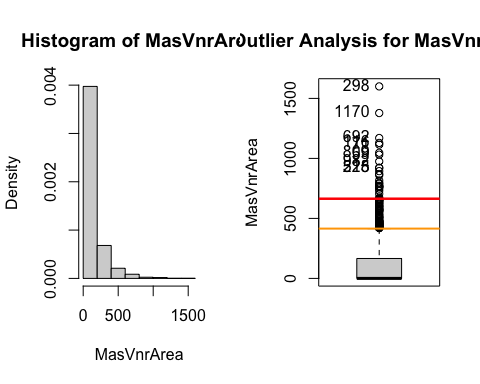
numeric\_description(train2$MasVnrArea, 10)# not normal distribution

## 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)

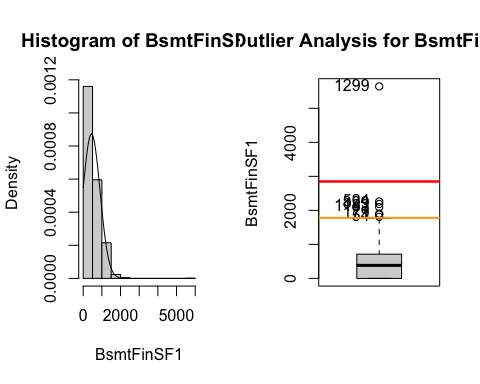
par(mfrow=c(1,2))  
numeric\_description(train2$BsmtFinSF1, 10)# not normal distribution

## 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)

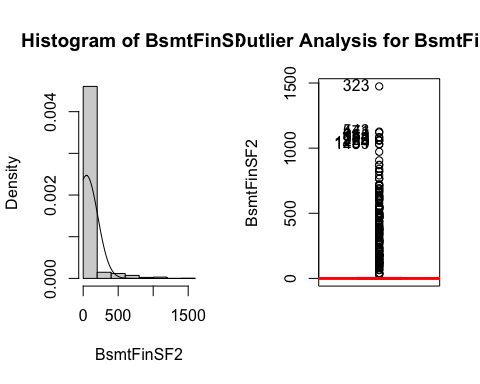
par(mfrow=c(1,2))  
numeric\_description(train2$BsmtFinSF2, 10) # not normal distribution

## 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)

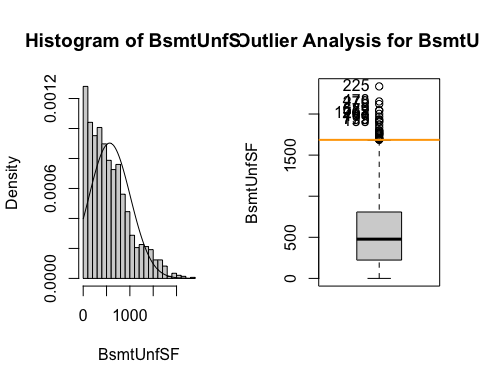
par(mfrow=c(1,2))  
numeric\_description(train2$BsmtUnfSF, 20) # not normal distribution

## 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)

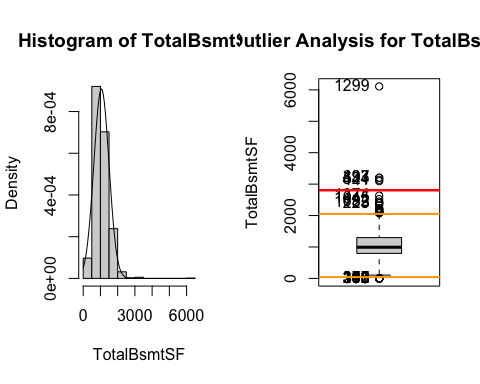
par(mfrow=c(1,2))  
numeric\_description(train2$TotalBsmtSF, 10)# not normal distribution

## 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)

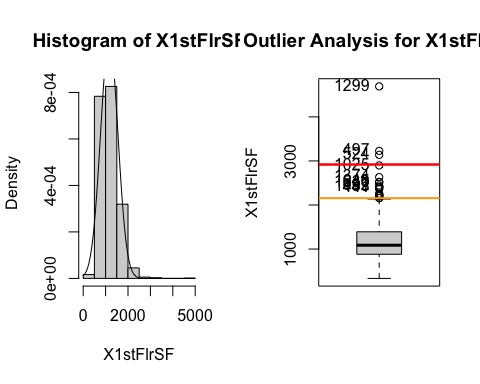
par(mfrow=c(1,2))  
numeric\_description(train2$X1stFlrSF, 10) # not normal distribution

## 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)

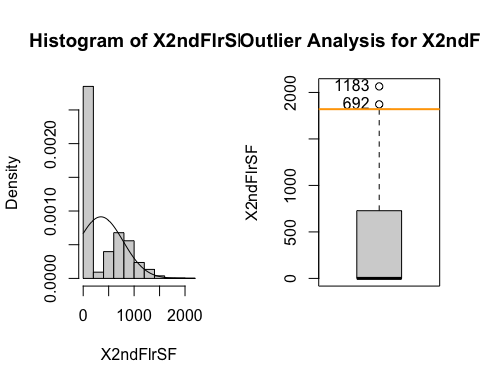
par(mfrow=c(1,2))  
numeric\_description(train2$X2ndFlrSF, 10)# not normal distribution

## 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)

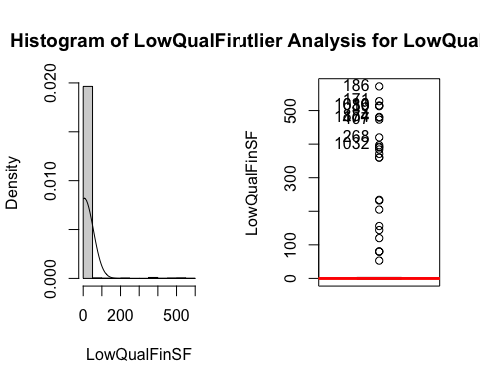
par(mfrow=c(1,2))  
numeric\_description(train2$LowQualFinSF, 10) # not normal distribution

## 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)

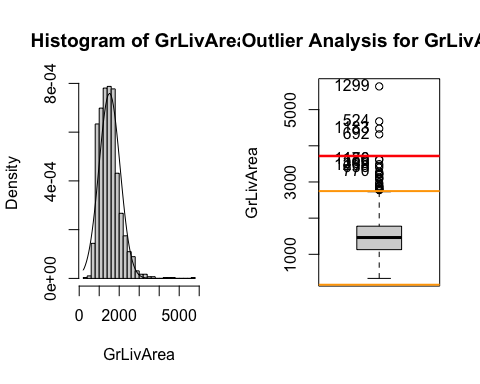
par(mfrow=c(1,2))  
numeric\_description(train2$GrLivArea, 20) # not normal distribution

## 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")

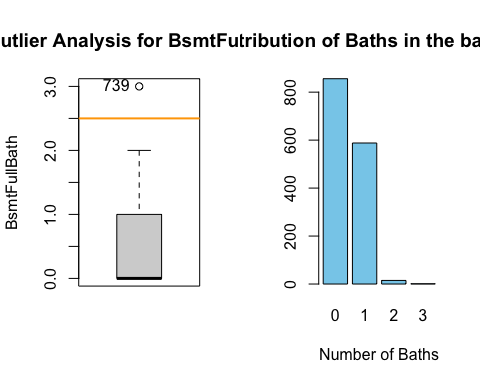
## Number of mild outliers: 1

## 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")



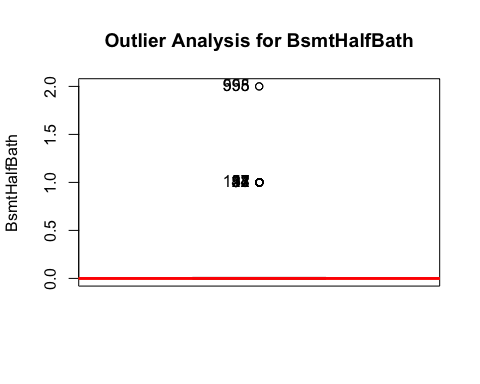
### BsmtHalfBath - # half baths in the basement (int)

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

Gráfico, Gráfico de barras

Descripción generada automáticamente

## 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

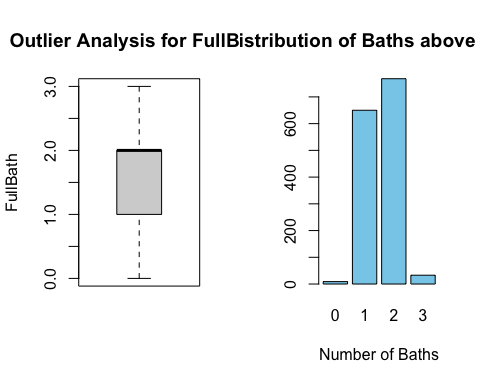
train2 <- update\_outliers\_count(train2, "FullBath")  
table(train2$FullBath)

##   
## 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

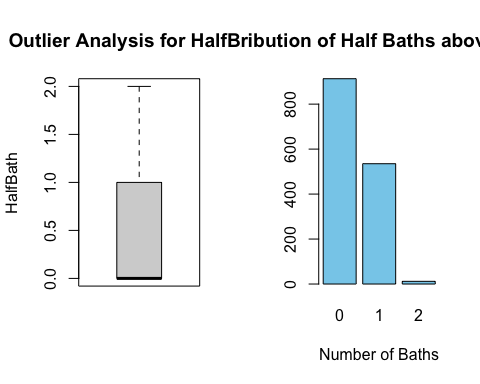
train2 <- update\_outliers\_count(train2, "HalfBath")  
table(train2$HalfBath)

##   
## 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

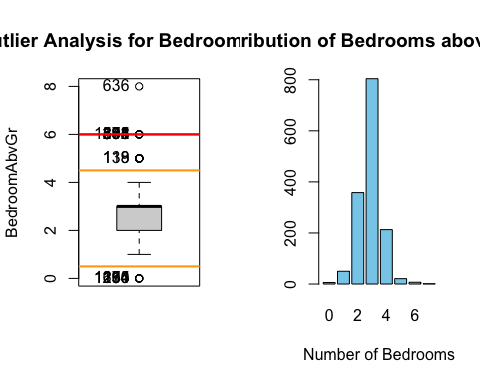
## Number of severe outliers: 1

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

sum(is.na(train2$BedroomAbvGr))

## [1] 0

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



### KitchenAbvGr - Kitchens above grade (int)

summary(train2$KitchenAbvGr)

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

analyze\_outliers(train2,"KitchenAbvGr")

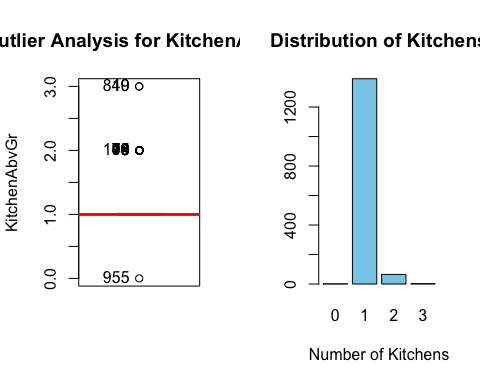
## Number of mild outliers: 68

## Number of severe outliers: 68

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

summary(train2$TotRmsAbvGrd)

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

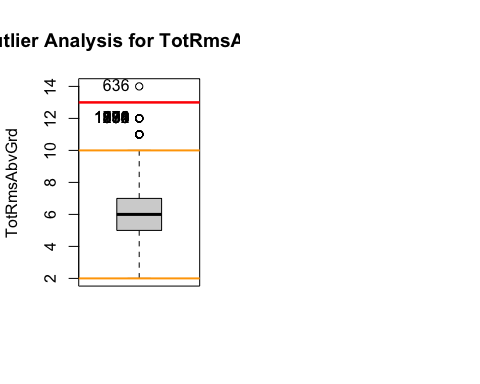
analyze\_outliers(train2,"TotRmsAbvGrd")

## 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

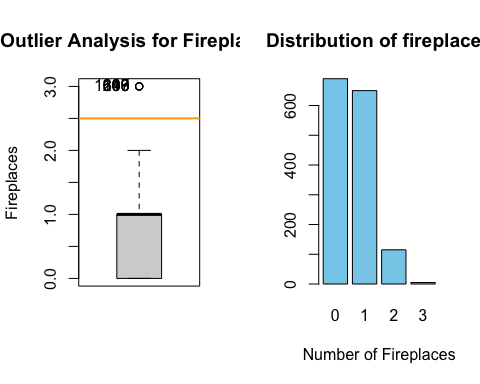
par(mfrow=c(1,2))  
#table(train2$Fireplaces)  
sum(is.na(train2$Fireplaces))

## [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

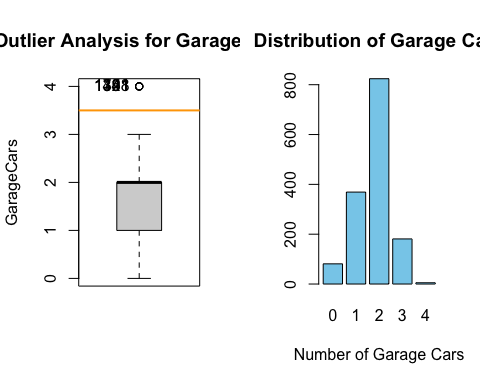
## [1] 0

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)

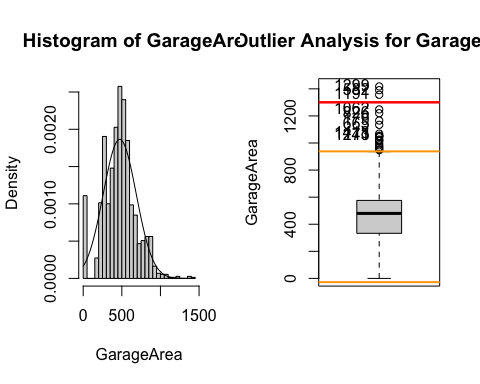
par(mfrow=c(1,2))  
numeric\_description(train2$GarageArea, 30) # not normal distribution

## 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)

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

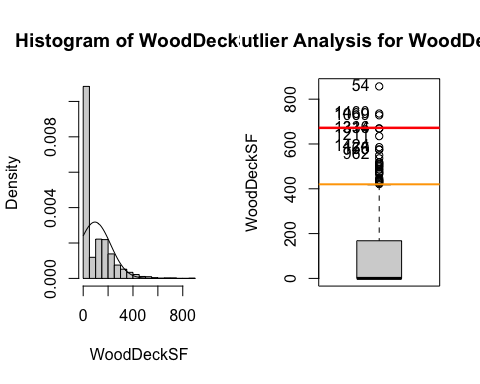
par(mfrow=c(1,2))  
numeric\_description(train2$WoodDeckSF, 30) # not normal distribution

## 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



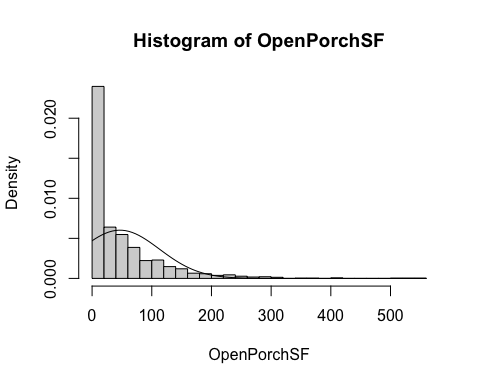
## Number of severe outliers: 3

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

Diagrama

Descripción generada automáticamente

##   
## 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

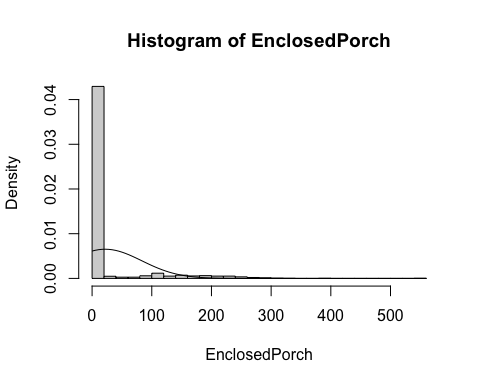
train2 <- update\_outliers\_count(train2, "OpenPorchSF")  
#table(train2$OpenPorchSF) # very centered data in 0  
train2$OpenPorch\_binary<-0  
train2$OpenPorch\_binary[which(train2$OpenPorchSF!=0)]<-1  
train2$OpenPorch\_binary<-as.factor(train2$OpenPorch\_binary)  
table(train2$OpenPorch\_binary)

##   
## 0 1   
## 656 804

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

numeric\_description(train2$EnclosedPorch, 30)

## Count of missing values: 0

Diagrama, Esquemático

Descripción generada automáticamente

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

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)

numeric\_description(train2$X3SsnPorch, 30)

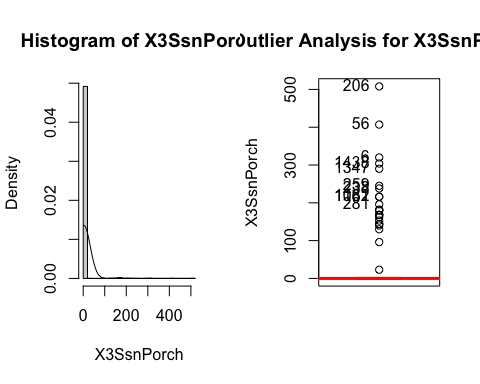
## Count of missing values: 0

##   
## 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)

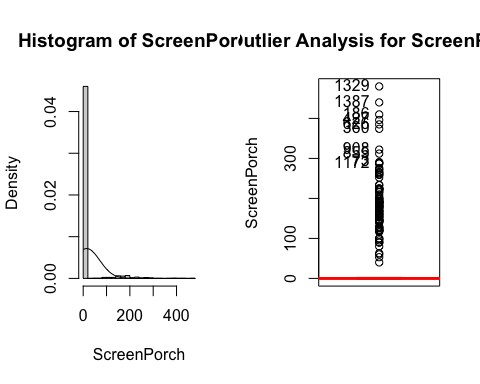
par(mfrow=c(1,2))  
numeric\_description(train2$ScreenPorch, 30) # not normally distributed

## 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)

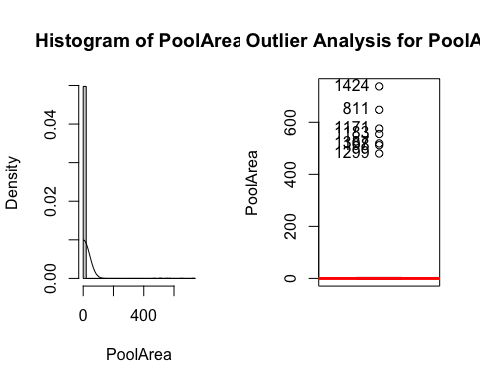
numeric\_description(train2$PoolArea, 30)# not normally distributed

## 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)

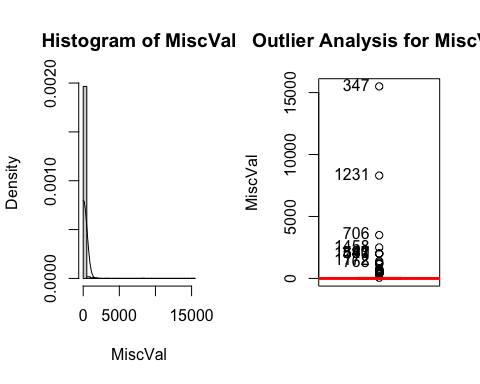
par(mfrow=c(1,2))  
numeric\_description(train2$MiscVal, 30) # not normally distributed

## 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



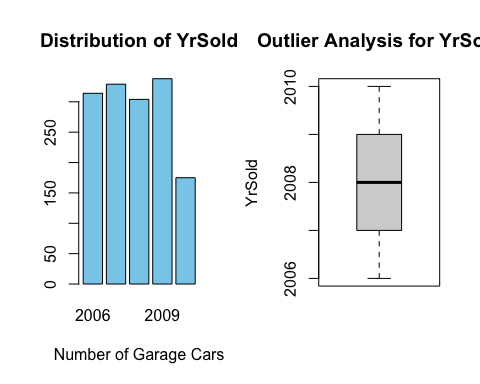
## Number of severe outliers: 52

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

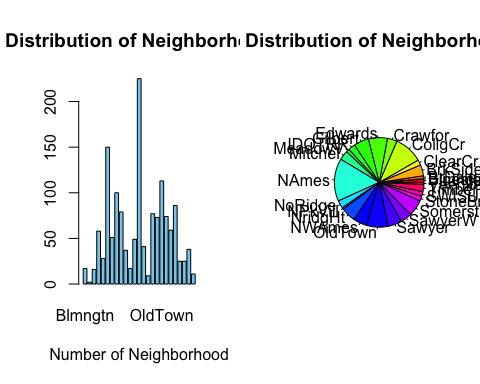
Categorical variables are analyzed.

### 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")Gráfico, Histograma

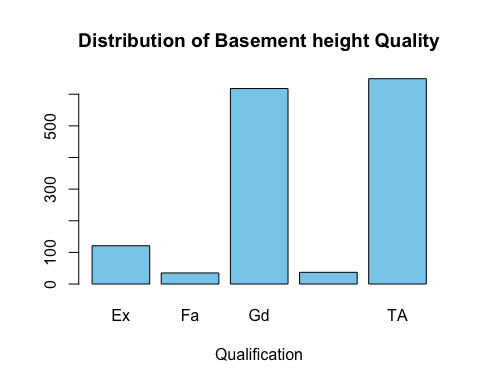
Descripción generada automáticamente

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")

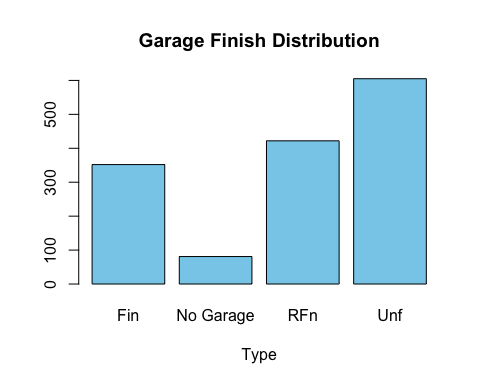


GarageFinish - Interior finish of the garage

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

## [1] 0

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

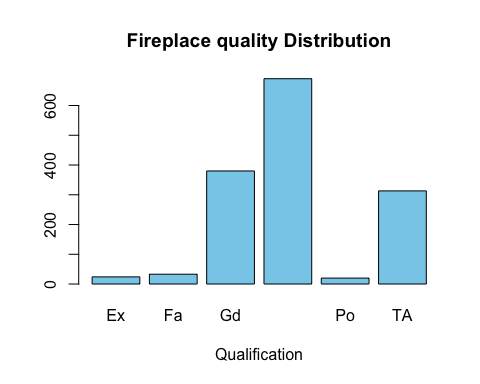


FireplaceQu - Fireplace quality

sum(is.na(train2$FireplaceQu))

## [1] 0

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

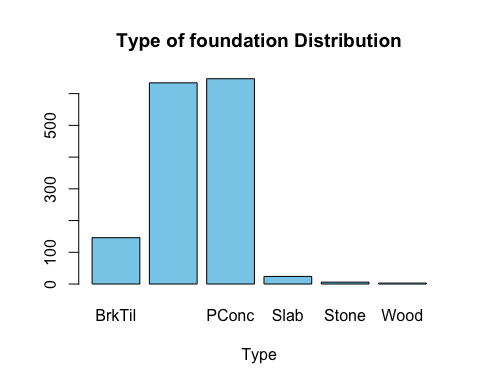


Foundation - Type of foundation

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

## [1] 0

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

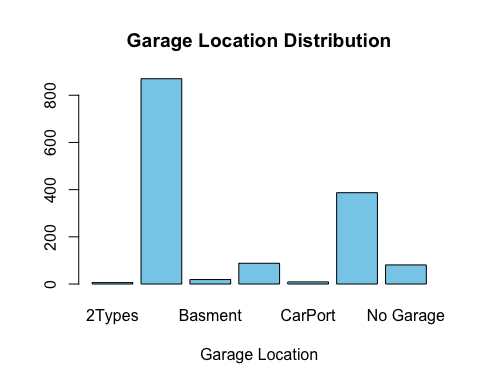


GarageType - Garage location

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 = "Garage Location",col = "skyblue")

Gráfico, Gráfico de barras

Descripción generada automáticamente

## 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  
## 5 5 LotFrontage MasVnrArea

… … .. … . .. .

imputed\_data<-complete(mice\_imp)

We proceed to validate the imputation.

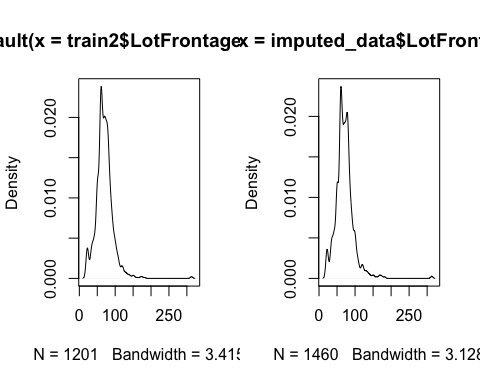
# LotFrontage validation  
summary(imputed\_data$LotFrontage)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 21 60 70 71 80 313

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))



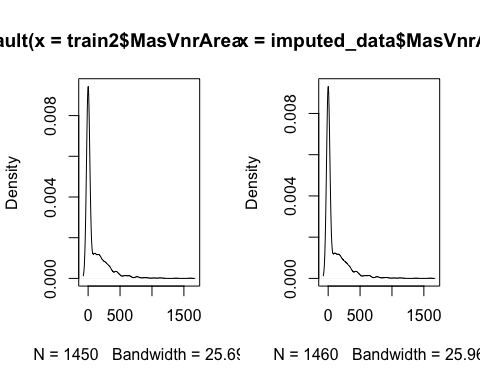
Checking the results we can conclude that the imputation doesn’t have a significant change in the density nor the summary.

# MasVnrArea validation  
summary(imputed\_data$MasVnrArea)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0 0.0 0.0 104.1 166.0 1600.0

summary(train2$MasVnrArea)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.0 0.0 0.0 103.2 164.0 1600.0 10  
plot(density(train2$MasVnrArea,na.rm=TRUE))  
plot(density(imputed\_data$MasVnrArea,na.rm=TRUE))



# 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)

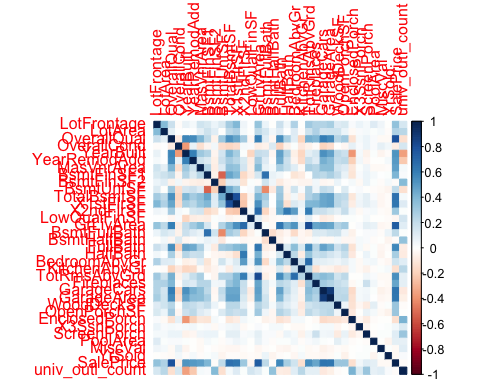
## [1] 10

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

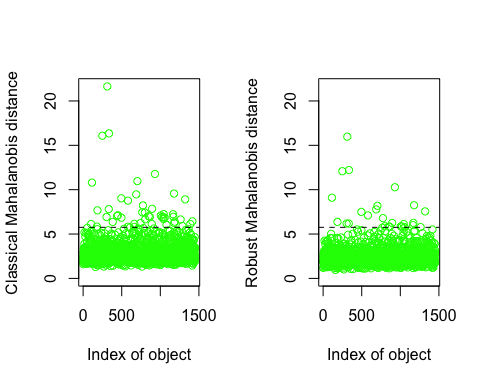
## Multivariate outliers

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")



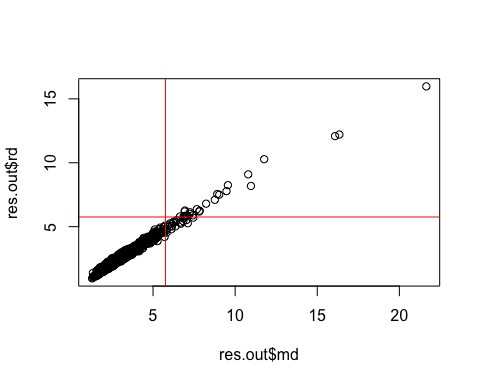
which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff))

## 114 186 250 305 314 333 336 496 584 692 707 779 804 855 869 899   
## 114 186 249 304 313 332 335 495 581 688 703 775 800 850 864 894   
## 935 1012 1025 1049 1062 1170 1174 1183 1269 1329   
## 930 1007 1020 1044 1057 1165 1169 1178 1263 1322

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

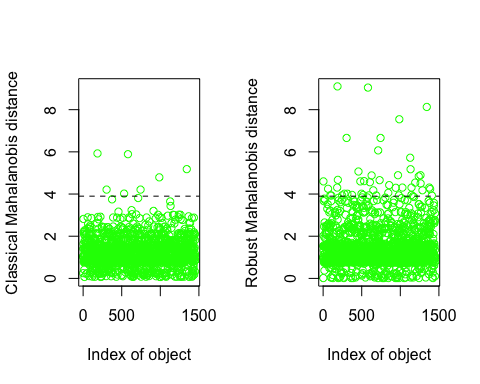
## [1] 26

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")

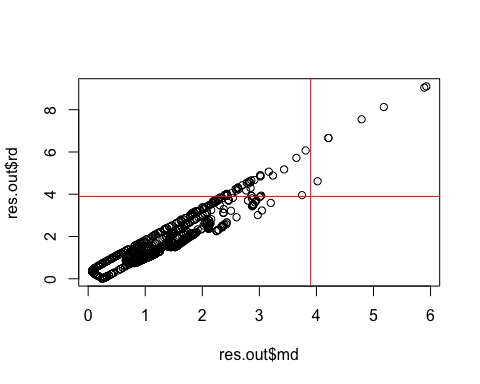


# take off m outliers  
#imputed\_data <- imputed\_data[-which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff)),]

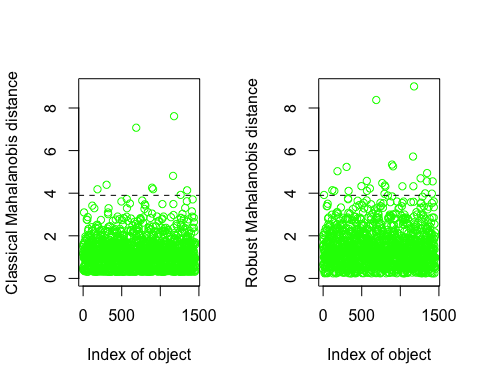
# 2. Generally the houses built latetly are the ones with most quality  
res.out <- Moutlier(imputed\_data[,c("YearBuilt","OverallQual")], quantile = 0.9995, col="green")



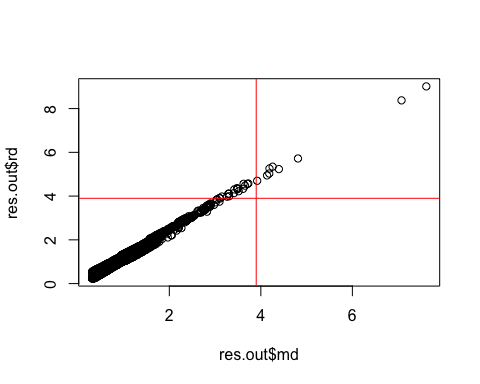
which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff))  
par(mfrow=c(1,1))  
plot( res.out$md, res.out$rd )  
abline(h=res.out$cutoff, col="red")  
abline(v=res.out$cutoff, col="red")



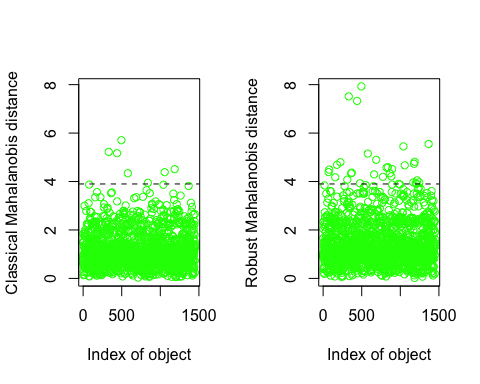
# 3. Generally the houses with more area are the ones with more rooms  
res.out <- Moutlier(imputed\_data[,c("GrLivArea","TotRmsAbvGrd")], quantile = 0.9995, col="green")



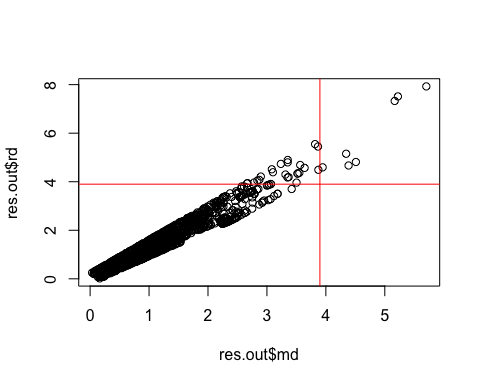
which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff))  
par(mfrow=c(1,1))  
plot( res.out$md, res.out$rd )  
abline(h=res.out$cutoff, col="red")  
abline(v=res.out$cutoff, col="red")



# 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")

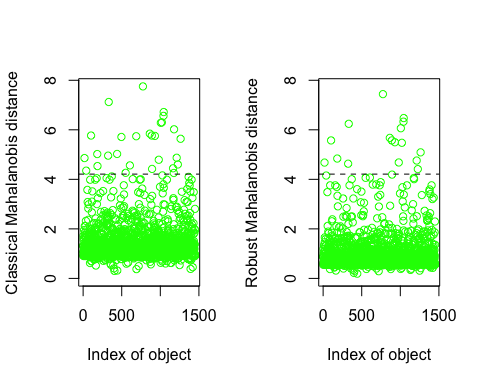


which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff))  
par(mfrow=c(1,1))  
plot( res.out$md, res.out$rd )  
abline(h=res.out$cutoff, col="red")  
abline(v=res.out$cutoff, col="red")

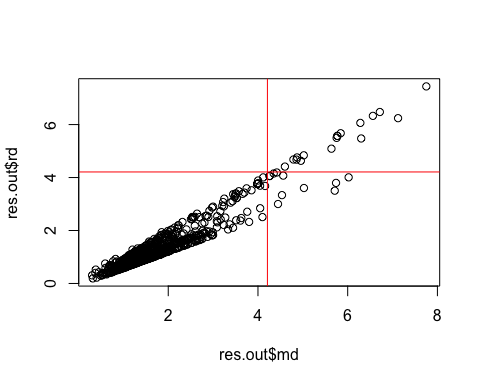


# 5. Moutliers about total areas zones  
res.out <- Moutlier(imputed\_data[,c("X1stFlrSF","TotalBsmtSF","GrLivArea")], quantile = 0.9995, col="green") #GarageArea, si se lo sumo hay menos outliers

## Warning in covMcd(X): The covariance matrix has become singular during  
## the iterations of the MCD algorithm.  
## There are 571 observations (in the entire dataset of 1452 obs.) lying  
## on the plane with equation 0.70711 (x\_i1-m\_1) + -0.70711 (x\_i2-m\_2) +  
## 3.3089e-17 (x\_i3-m\_3) = 0 with (m\_1,m\_2) the mean of these  
## observations.



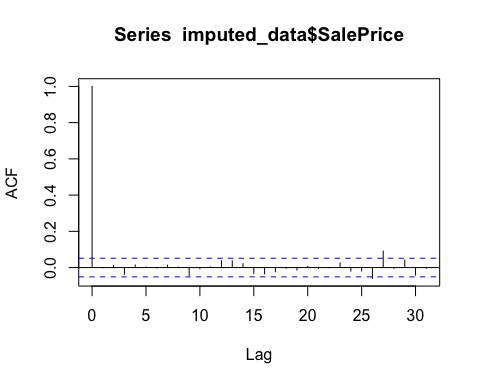
which((res.out$md > res.out$cutoff)&(res.out$rd > res.out$cutoff))  
par(mfrow=c(1,1))  
plot( res.out$md, res.out$rd )  
abline(h=res.out$cutoff, col="red")  
abline(v=res.out$cutoff, col="red")



# 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)



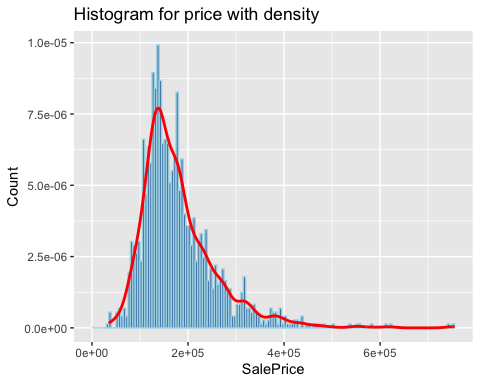
ks.test(imputed\_data$SalePrice, 'pnorm', mean(imputed\_data$SalePrice), sd(imputed\_data$SalePrice))

## 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))  
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))

…

See the building of the model in annex.2

We stop here, the model we end up with is : (summary in annex)

m15<- lm(log(SalePrice) ~OverallQual+log(GrLivArea)+GarageCars+GarageArea+TotalBsmtSF+YearBuilt+YearRemodAdd+Fireplaces+BsmtFinSF1+LotFrontage+Woo

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 -962.5599  
## m12 10 -1022.0880  
## m13 11 -1118.0725  
## m14 12 -1149.2444  
## m15 13 -1159.8366

summary(m15)$r.squared

## [1] 0.884595

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

## [1] 8.665136

vif(m15) # there are not very correlated predictors

## OverallQual log(GrLivArea) GarageCars GarageArea TotalBsmtSF   
## 2.952284 2.122590 5.508182 5.169565 1.865130   
## YearBuilt YearRemodAdd Fireplaces BsmtFinSF1 LotFrontage   
## 2.010514 1.742853 1.456881 1.378455 1.214537   
## WoodDeckSF   
## 1.138120

We can also use the automatic forward stepwise step() and see which are the variables that will create a model with higher accuracy. Summary in annex.3

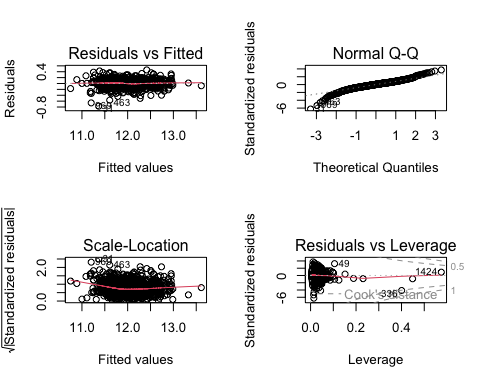
mod.fow <- stats::step(lm(SalePrice~. , data = train\_new), trace = FALSE, direction = "forward")

summary(mod.fow)

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] 0.904289

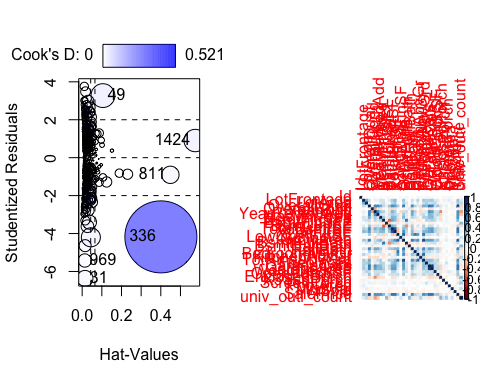
par(mfrow=c(2,2))  
plot(n00)



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

## StudRes Hat CookD  
## 811 -0.9038077 0.44874309 0.03023104  
## 31 -6.3654784 0.01427126 0.02564555  
## 49 3.2729933 0.10552683 0.05689056  
## 336 -4.1758594 0.40040864 0.52070761  
## 969 -5.4210209 0.01347073 0.01773338  
## 1424 0.9038077 0.57499765 0.05024399

corrplot(descrCor, method = 'color')



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

## [1] 10.44812

vif(n00)

## LotArea OverallQual OverallCond YearBuilt MasVnrArea BsmtFinSF1   
## 1.202712 3.298939 1.253192 2.831705 1.438244 5.360580   
## BsmtFinSF2 BsmtUnfSF X1stFlrSF X2ndFlrSF BsmtFullBath FullBath   
## 1.532288 4.335038 4.484607 4.567232 1.980196 2.888001   
## HalfBath BedroomAbvGr KitchenAbvGr Fireplaces GarageCars OpenPorchSF   
## 2.247048 1.670234 1.287219 1.529912 1.966977 1.187960   
## X3SsnPorch ScreenPorch PoolArea   
## 1.014984 1.063388 1.019185

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 -1159.837  
## 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 18.515   
## 2 1001 16.197 3 2.3185 47.763 < 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] 0.884595

We will proceed with the n0 model.

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)  
  
# 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 \*\*\*  
## ---

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 \*\*  
## ---

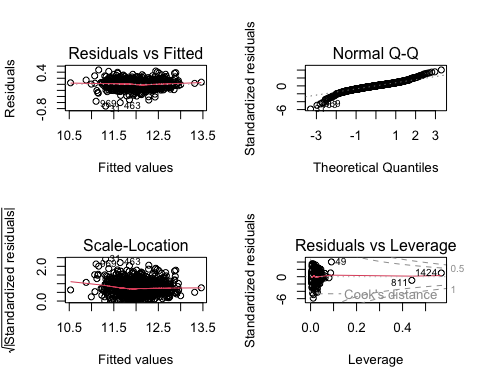
v1<- lm(log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + BsmtFinSF2+ log(X1stFlrSF)+ X2ndFlrSF +  
 BsmtFullBath + KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea, data = train\_new)  
  
  
# X2stFlrSF  
  
boxTidwell(log(SalePrice) ~ I(X2ndFlrSF+0.01),~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

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.

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. All the steps of the variable selection in the annex.

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)$r.squared

## [1] 0.9208199

anova(v1,b0)  
## 2 975 12.703 26 2.5416 7.5028 < 2.2e-16 \*\*\*

AIC(v1,b0)

## df AIC  
## v1 16 -1351.285  
## b0 42 -1484.587

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

## [1] 12.62944

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

## Neighborhood 44.452433 24 1.082259

........

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

## df AIC  
## b7 34 -1440.185  
## b71 98 -1457.690  
## b3 28 -1441.765  
## b31 86 -1438.305  
## b2 25 -1433.435  
## b21 88 -1394.021  
## b1 21 -1387.641  
## b11 59 -1396.775

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 21677.13  
## 2 Model with test\_new sample 0.92 23843.28

## Study the presence of a priori influential data observations

In this case, 26 a priori values where found.

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

## [1] 0.0265748

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

## [1] 13

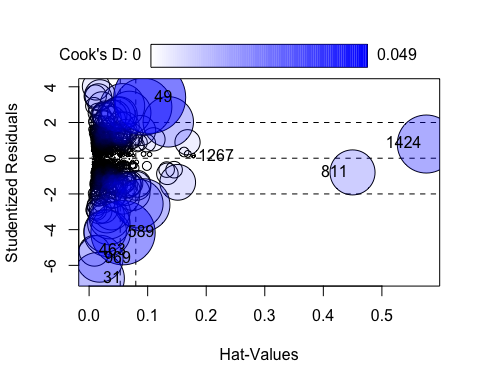
## Study the presence of a posteriori influential values

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

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

## [1] 0.06274558

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



## StudRes Hat CookD  
## 811 -0.7946716 0.449851043 1.913207e-02  
## 31 -6.7300688 0.015716535 2.563801e-02  
## 49 3.4030501 0.103417853 4.895046e-02  
## 589 -4.1720012 0.057915832 3.898411e-02  
## 1267 0.0916685 0.178297696 6.759947e-05  
## 969 -5.6273650 0.017938865 2.077977e-02  
## 1424 0.7946716 0.575851845 3.176630e-02  
## 463 -5.1829520 0.009131944 8.935666e-03

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

## [1] 26

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

## [1] 29

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

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

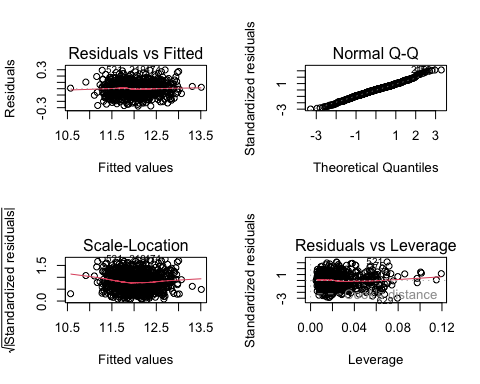
train\_new<-train\_new[-llremreg,]  
  
final\_modelp <- lm(log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond + YearBuilt +  
 BsmtFinSF1 + log(X1stFlrSF)+ X2ndFlrSF +  
 BsmtFullBath + FullBath + TotRmsAbvGrd+  
 KitchenAbvGr + Fireplaces + GarageCars +  
 ScreenPorch + PoolArea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

##   
## Call:  
## lm(formula = log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual +   
## OverallCond + YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF +   
## BsmtFullBath + FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces +   
## GarageCars + ScreenPorch + PoolArea + ExterQual + BsmtQual +   
## KitchenQual + GarageType, data = train\_new)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.272598 -0.059962 0.002771 0.058528 0.293911   
##   
## Coefficients: (1 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.502e+00 4.583e-01 3.277 0.001086 \*\*   
## LotFrontage -7.357e-05 1.826e-04 -0.403 0.687211   
## log(LotArea) 9.798e-02 8.922e-03 10.982 < 2e-16 \*\*\*  
## OverallQual 6.076e-02 4.334e-03 14.021 < 2e-16 \*\*\*  
## OverallCond 5.210e-02 3.191e-03 16.327 < 2e-16 \*\*\*  
## YearBuilt 3.112e-03 2.027e-04 15.349 < 2e-16 \*\*\*  
## BsmtFinSF1 7.584e-05 1.024e-05 7.405 2.88e-13 \*\*\*  
## log(X1stFlrSF) 3.971e-01 2.018e-02 19.675 < 2e-16 \*\*\*  
## X2ndFlrSF 2.434e-04 1.476e-05 16.483 < 2e-16 \*\*\*  
## BsmtFullBath 2.121e-02 7.982e-03 2.657 0.008018 \*\*   
## FullBath 3.009e-02 8.771e-03 3.431 0.000628 \*\*\*  
## TotRmsAbvGrd -1.390e-03 3.804e-03 -0.365 0.714878   
## KitchenAbvGr -7.878e-02 1.784e-02 -4.415 1.13e-05 \*\*\*  
## Fireplaces 2.157e-02 5.767e-03 3.741 0.000194 \*\*\*  
## GarageCars 4.571e-02 7.162e-03 6.382 2.72e-10 \*\*\*  
## ScreenPorch 2.866e-04 6.017e-05 4.763 2.21e-06 \*\*\*  
## PoolArea NA NA NA NA   
## ExterQualFa -6.212e-02 4.413e-02 -1.408 0.159554   
## ExterQualGd -8.254e-03 2.110e-02 -0.391 0.695737   
## ExterQualTA -4.820e-02 2.360e-02 -2.042 0.041376 \*   
## BsmtQualFa -1.709e-01 2.811e-02 -6.079 1.74e-09 \*\*\*  
## BsmtQualGd -9.242e-02 1.569e-02 -5.890 5.34e-09 \*\*\*  
## BsmtQualNo Basement -2.206e-01 2.863e-02 -7.704 3.28e-14 \*\*\*  
## BsmtQualTA -9.954e-02 1.848e-02 -5.385 9.11e-08 \*\*\*  
## KitchenQualFa -1.042e-01 3.004e-02 -3.470 0.000544 \*\*\*  
## KitchenQualGd -6.528e-02 1.575e-02 -4.146 3.68e-05 \*\*\*  
## KitchenQualTA -9.331e-02 1.766e-02 -5.284 1.57e-07 \*\*\*  
## GarageTypeAttchd 8.218e-02 5.034e-02 1.632 0.102918   
## GarageTypeBasment 1.147e-01 5.766e-02 1.990 0.046866 \*   
## GarageTypeBuiltIn 8.682e-02 5.225e-02 1.662 0.096930 .   
## GarageTypeCarPort 5.657e-03 6.368e-02 0.089 0.929235   
## GarageTypeDetchd 9.561e-02 5.010e-02 1.908 0.056657 .   
## GarageTypeNo Garage 9.750e-02 5.407e-02 1.803 0.071664 .   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.09572 on 956 degrees of freedom  
## Multiple R-squared: 0.9407, Adjusted R-squared: 0.9387   
## F-statistic: 488.8 on 31 and 956 DF, p-value: < 2.2e-16

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

## Start: AIC=-4448.23  
## log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + PoolArea + ExterQual + BsmtQual + KitchenQual +   
## GarageType  
##   
##   
## Step: AIC=-4448.23  
## 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 0.0970 8.8560 -4478.7  
## - ExterQual 3 0.1195 8.8785 -4455.5  
## - TotRmsAbvGrd 1 0.0012 8.7602 -4455.0  
## - LotFrontage 1 0.0015 8.7605 -4455.0  
## <none> 8.7590 -4448.2  
## - BsmtFullBath 1 0.0647 8.8237 -4447.9  
## - FullBath 1 0.1078 8.8668 -4443.0  
## - Fireplaces 1 0.1282 8.8872 -4440.8  
## - KitchenQual 3 0.2602 9.0192 -4440.0  
## - KitchenAbvGr 1 0.1786 8.9376 -4435.2  
## - ScreenPorch 1 0.2078 8.9668 -4432.0  
## - GarageCars 1 0.3732 9.1322 -4413.9  
## - BsmtQual 4 0.6597 9.4187 -4404.1  
## - BsmtFinSF1 1 0.5024 9.2614 -4400.0  
## - log(LotArea) 1 1.1050 9.8640 -4337.7  
## - OverallQual 1 1.8011 10.5601 -4270.4  
## - YearBuilt 1 2.1585 10.9175 -4237.5  
## - OverallCond 1 2.4424 11.2014 -4212.1  
## - X2ndFlrSF 1 2.4893 11.2483 -4208.0  
## - log(X1stFlrSF) 1 3.5466 12.3056 -4119.2  
##   
## Step: AIC=-4478.72  
## 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 0.1180 8.9740 -4486.3  
## - TotRmsAbvGrd 1 0.0004 8.8564 -4485.6  
## - LotFrontage 1 0.0017 8.8577 -4485.4  
## <none> 8.8560 -4478.7  
## - BsmtFullBath 1 0.0660 8.9220 -4478.3  
## - FullBath 1 0.1055 8.9615 -4473.9  
## - KitchenQual 3 0.2560 9.1120 -4471.3  
## - Fireplaces 1 0.1304 8.9864 -4471.2  
## - ScreenPorch 1 0.2072 9.0632 -4462.8  
## - KitchenAbvGr 1 0.2282 9.0842 -4460.5  
## - BsmtQual 4 0.6754 9.5314 -4433.7  
## - BsmtFinSF1 1 0.4906 9.3466 -4432.4  
## - GarageCars 1 0.5046 9.3606 -4430.9  
## - log(LotArea) 1 1.0616 9.9177 -4373.8  
## - OverallQual 1 1.8824 10.7384 -4295.2  
## - YearBuilt 1 2.4228 11.2788 -4246.7  
## - OverallCond 1 2.4722 11.3282 -4242.4  
## - X2ndFlrSF 1 2.5824 11.4384 -4232.8  
## - log(X1stFlrSF) 1 3.5833 12.4393 -4149.9  
##   
## Step: AIC=-4486.33  
## log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual + OverallCond +   
## YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces + GarageCars +   
## ScreenPorch + BsmtQual + KitchenQual  
##   
## Df Sum of Sq RSS AIC  
## - TotRmsAbvGrd 1 0.0008 8.9748 -4493.1  
## - LotFrontage 1 0.0043 8.9783 -4492.8  
## <none> 8.9740 -4486.3  
## - BsmtFullBath 1 0.0720 9.0459 -4485.3  
## - FullBath 1 0.1040 9.0780 -4481.8  
## - Fireplaces 1 0.1165 9.0905 -4480.5  
## - ScreenPorch 1 0.2040 9.1780 -4471.0  
## - KitchenAbvGr 1 0.2212 9.1952 -4469.2  
## - KitchenQual 3 0.4644 9.4384 -4457.2  
## - BsmtFinSF1 1 0.4574 9.4314 -4444.1  
## - GarageCars 1 0.5083 9.4823 -4438.8  
## - BsmtQual 4 0.7240 9.6980 -4437.3  
## - log(LotArea) 1 1.0713 10.0453 -4381.8  
## - OverallQual 1 2.4211 11.3951 -4257.2  
## - OverallCond 1 2.5373 11.5113 -4247.2  
## - X2ndFlrSF 1 2.6219 11.5959 -4240.0  
## - YearBuilt 1 2.6592 11.6332 -4236.8  
## - log(X1stFlrSF) 1 3.6775 12.6515 -4153.9  
##   
## Step: AIC=-4493.14  
## 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 0.0044 8.9793 -4499.5  
## <none> 8.9748 -4493.1  
## - BsmtFullBath 1 0.0717 9.0465 -4492.2  
## - FullBath 1 0.1032 9.0780 -4488.7  
## - Fireplaces 1 0.1165 9.0913 -4487.3  
## - ScreenPorch 1 0.2054 9.1803 -4477.7  
## - KitchenAbvGr 1 0.2399 9.2148 -4474.0  
## - KitchenQual 3 0.4643 9.4391 -4464.0  
## - BsmtFinSF1 1 0.4744 9.4492 -4449.1  
## - GarageCars 1 0.5077 9.4826 -4445.7  
## - BsmtQual 4 0.7286 9.7034 -4443.6  
## - log(LotArea) 1 1.0715 10.0463 -4388.6  
## - OverallQual 1 2.4247 11.3995 -4263.8  
## - OverallCond 1 2.5375 11.5124 -4254.0  
## - YearBuilt 1 2.6595 11.6343 -4243.6  
## - X2ndFlrSF 1 4.8123 13.7871 -4075.9  
## - log(X1stFlrSF) 1 4.8582 13.8330 -4072.6  
##   
## Step: AIC=-4499.54  
## log(SalePrice) ~ log(LotArea) + OverallQual + OverallCond + YearBuilt +   
## BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF + BsmtFullBath +   
## FullBath + KitchenAbvGr + Fireplaces + GarageCars + ScreenPorch +   
## BsmtQual + KitchenQual  
##   
## Df Sum of Sq RSS AIC  
## <none> 8.9793 -4499.5  
## - BsmtFullBath 1 0.0736 9.0529 -4498.4  
## - FullBath 1 0.1030 9.0823 -4495.2  
## - Fireplaces 1 0.1162 9.0955 -4493.7  
## - ScreenPorch 1 0.2060 9.1853 -4484.0  
## - KitchenAbvGr 1 0.2392 9.2185 -4480.5  
## - KitchenQual 3 0.4681 9.4474 -4470.0  
## - BsmtFinSF1 1 0.4753 9.4546 -4455.5  
## - GarageCars 1 0.5034 9.4826 -4452.6  
## - BsmtQual 4 0.7242 9.7035 -4450.5  
## - log(LotArea) 1 1.4681 10.4474 -4356.8  
## - OverallQual 1 2.4310 11.4103 -4269.7  
## - OverallCond 1 2.5331 11.5124 -4260.9  
## - YearBuilt 1 2.6583 11.6376 -4250.2  
## - X2ndFlrSF 1 4.8079 13.7872 -4082.8  
## - log(X1stFlrSF) 1 4.8760 13.8552 -4077.9

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



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

## Variable Original\_coefficient  
## (Intercept) (Intercept) 1.718154e+00  
## log(LotArea) log(LotArea) 9.666169e-02  
## OverallQual OverallQual 6.663446e-02  
## OverallCond OverallCond 5.278106e-02  
## YearBuilt YearBuilt 3.051238e-03  
## BsmtFinSF1 BsmtFinSF1 7.835193e-05  
## log(X1stFlrSF) log(X1stFlrSF) 3.895264e-01  
## X2ndFlrSF X2ndFlrSF 2.326831e-04  
## BsmtFullBath BsmtFullBath 2.270556e-02  
## FullBath FullBath 2.882518e-02  
## KitchenAbvGr KitchenAbvGr -6.954690e-02  
## Fireplaces Fireplaces 2.475745e-02  
## GarageCars GarageCars 4.913981e-02  
## ScreenPorch ScreenPorch 2.941106e-04  
## BsmtQualFa BsmtQualFa -1.554858e-01  
## BsmtQualGd BsmtQualGd -9.711389e-02  
## BsmtQualNo Basement BsmtQualNo Basement -2.213174e-01  
## BsmtQualTA BsmtQualTA -1.068236e-01  
## KitchenQualFa KitchenQualFa -6.028988e-02  
## KitchenQualGd KitchenQualGd -4.242676e-02  
## KitchenQualTA KitchenQualTA -7.403817e-02  
## Filtered\_coefficient  
## (Intercept) 1.589139e+00  
## log(LotArea) 9.219227e-02  
## OverallQual 6.658310e-02  
## OverallCond 5.175633e-02  
## YearBuilt 3.126180e-03  
## BsmtFinSF1 7.241335e-05  
## log(X1stFlrSF) 3.934097e-01  
## X2ndFlrSF 2.401428e-04  
## BsmtFullBath 2.250707e-02  
## FullBath 2.895870e-02  
## KitchenAbvGr -8.620365e-02  
## Fireplaces 2.027016e-02  
## GarageCars 4.260217e-02  
## ScreenPorch 2.840820e-04  
## BsmtQualFa -1.771345e-01  
## BsmtQualGd -9.503964e-02  
## BsmtQualNo Basement -2.248469e-01  
## BsmtQualTA -1.070349e-01  
## KitchenQualFa -1.134850e-01  
## KitchenQualGd -6.608734e-02  
## KitchenQualTA -1.099259e-01

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

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

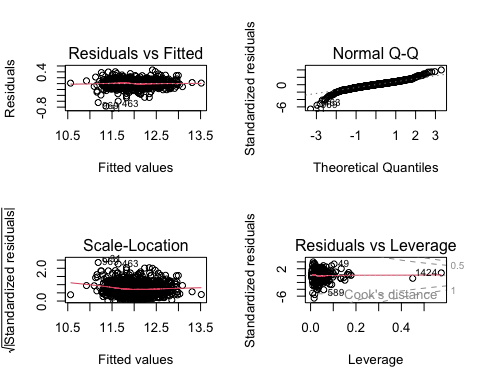
## Interpret the Model

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

summary(final\_model)

##   
## Call:  
## lm(formula = log(SalePrice) ~ LotFrontage + log(LotArea) + OverallQual +   
## OverallCond + YearBuilt + BsmtFinSF1 + log(X1stFlrSF) + X2ndFlrSF +   
## BsmtFullBath + FullBath + TotRmsAbvGrd + KitchenAbvGr + Fireplaces +   
## GarageCars + ScreenPorch + PoolArea + ExterQual + BsmtQual +   
## KitchenQual, data = train\_new)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.76659 -0.05956 0.00535 0.06457 0.46559   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.718e+00 5.072e-01 3.387 0.000734 \*\*\*  
## LotFrontage -3.113e-04 2.117e-04 -1.470 0.141898   
## log(LotArea) 9.666e-02 1.062e-02 9.102 < 2e-16 \*\*\*  
## OverallQual 6.663e-02 5.202e-03 12.810 < 2e-16 \*\*\*  
## OverallCond 5.278e-02 3.834e-03 13.767 < 2e-16 \*\*\*  
## YearBuilt 3.051e-03 2.243e-04 13.602 < 2e-16 \*\*\*  
## BsmtFinSF1 7.835e-05 1.243e-05 6.302 4.42e-10 \*\*\*  
## log(X1stFlrSF) 3.895e-01 2.403e-02 16.209 < 2e-16 \*\*\*  
## X2ndFlrSF 2.327e-04 1.741e-05 13.365 < 2e-16 \*\*\*  
## BsmtFullBath 2.271e-02 9.693e-03 2.342 0.019355 \*   
## FullBath 2.883e-02 1.059e-02 2.722 0.006600 \*\*   
## TotRmsAbvGrd -3.160e-04 4.576e-03 -0.069 0.944959   
## KitchenAbvGr -6.955e-02 2.014e-02 -3.453 0.000577 \*\*\*  
## Fireplaces 2.476e-02 6.926e-03 3.575 0.000368 \*\*\*  
## GarageCars 4.914e-02 6.940e-03 7.081 2.71e-12 \*\*\*  
## ScreenPorch 2.941e-04 7.345e-05 4.004 6.69e-05 \*\*\*  
## PoolArea 1.573e-04 1.211e-04 1.299 0.194184   
## ExterQualFa -6.596e-02 5.287e-02 -1.247 0.212540   
## ExterQualGd -2.264e-02 2.544e-02 -0.890 0.373592   
## ExterQualTA -5.343e-02 2.831e-02 -1.887 0.059446 .   
## BsmtQualFa -1.555e-01 3.374e-02 -4.609 4.58e-06 \*\*\*  
## BsmtQualGd -9.711e-02 1.902e-02 -5.105 3.98e-07 \*\*\*  
## BsmtQualNo Basement -2.213e-01 3.463e-02 -6.391 2.54e-10 \*\*\*  
## BsmtQualTA -1.068e-01 2.220e-02 -4.811 1.74e-06 \*\*\*  
## KitchenQualFa -6.029e-02 3.523e-02 -1.711 0.087313 .   
## KitchenQualGd -4.243e-02 1.877e-02 -2.261 0.023990 \*   
## KitchenQualTA -7.404e-02 2.101e-02 -3.523 0.000446 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1174 on 989 degrees of freedom  
## Multiple R-squared: 0.9151, Adjusted R-squared: 0.9129   
## F-statistic: 410 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   
## 1.718154e+00 -3.112644e-04 9.666169e-02 6.663446e-02   
## OverallCond YearBuilt BsmtFinSF1 log(X1stFlrSF)   
## 5.278106e-02 3.051238e-03 7.835193e-05 3.895264e-01   
## X2ndFlrSF BsmtFullBath FullBath TotRmsAbvGrd   
## 2.326831e-04 2.270556e-02 2.882518e-02 -3.159864e-04   
## KitchenAbvGr Fireplaces GarageCars ScreenPorch   
## -6.954690e-02 2.475745e-02 4.913981e-02 2.941106e-04   
## PoolArea ExterQualFa ExterQualGd ExterQualTA   
## 1.573204e-04 -6.595516e-02 -2.264172e-02 -5.342710e-02   
## BsmtQualFa BsmtQualGd BsmtQualNo Basement BsmtQualTA   
## -1.554858e-01 -9.711389e-02 -2.213174e-01 -1.068236e-01   
## KitchenQualFa KitchenQualGd KitchenQualTA   
## -6.028988e-02 -4.242676e-02 -7.403817e-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. T

he 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.

# 

# Test predictions

We compute the same preprocess to the test sample (annex) and we predict the target variable.

test<-read.delim("test.csv", sep=',')   
… … …

predictions <- predict(final\_model, imputed\_test)  
predictions <- exp(predictions)  
prediccion <- data.frame("id"=imputed\_test$Id, "SalePrice"=predictions)  
write.table(prediccion, file="SalePrice Prediction", sep="\t", col.names=TRUE, row.names=FALSE)

## Annex

1. DATA OVERVIEW

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   
## $ 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   
## $ YearRemodAdd : int 2003 1976 2002 1970 2000 1995 2005 1973 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   
## $ 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   
## $ SaleType : chr "WD" "WD" "WD" "WD" ...  
## $ SaleCondition: chr "Normal" "Normal" "Normal" "Abnorml" ...  
## $ SalePrice : int 208500 181500 223500 140000 250000 143000

1. Model building manual stepwise

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)

#Chosen model

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   
## -0.84496 -0.06663 0.00823 0.07515 0.46143   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 1.303e-01 5.503e-01 0.237 0.812866   
## OverallQual 8.326e-02 5.230e-03 15.920 < 2e-16 \*\*\*  
## log(GrLivArea) 3.626e-01 1.905e-02 19.028 < 2e-16 \*\*\*  
## GarageCars 2.364e-02 1.310e-02 1.804 0.071500 .   
## GarageArea 1.313e-04 4.493e-05 2.922 0.003554 \*\*   
## TotalBsmtSF 1.120e-04 1.438e-05 7.789 1.68e-14 \*\*\*  
## YearBuilt 1.574e-03 2.001e-04 7.868 9.28e-15 \*\*\*  
## YearRemodAdd 2.649e-03 2.735e-04 9.686 < 2e-16 \*\*\*  
## Fireplaces 4.490e-02 7.918e-03 5.671 1.86e-08 \*\*\*  
## BsmtFinSF1 1.134e-04 1.158e-05 9.793 < 2e-16 \*\*\*  
## LotFrontage 1.120e-03 1.954e-04 5.735 1.29e-08 \*\*\*  
## WoodDeckSF 1.287e-04 3.636e-05 3.538 0.000421 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.1358 on 1004 degrees of freedom  
## Multiple R-squared: 0.8846, Adjusted R-squared: 0.8833   
## F-statistic: 699.6 on 11 and 1004 DF, p-value: < 2.2e-16

1. Summary automatic stepwise

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   
## -92124 -10948 445 10428 168195   
##   
## Coefficients: (4 not defined because of singularities)  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 2.222e+06 1.241e+06 1.790 0.073734 .   
## Id 1.171e+00 1.886e+00 0.621 0.534911   
## LotFrontage 9.402e+00 4.640e+01 0.203 0.839467   
## LotArea 5.625e-01 1.464e-01 3.843 0.000130 \*\*\*  
## OverallQual 8.795e+03 1.187e+03 7.407 2.96e-13 \*\*\*  
## OverallCond 6.568e+03 9.357e+02 7.020 4.36e-12 \*\*\*  
## YearBuilt 4.333e+02 9.739e+01 4.450 9.67e-06 \*\*\*  
## YearRemodAdd 9.211e+01 6.169e+01 1.493 0.135746   
## MasVnrArea 1.868e+01 5.630e+00 3.319 0.000939 \*\*\*  
## BsmtFinSF1 4.335e+01 5.906e+00 7.340 4.75e-13 \*\*\*  
## BsmtFinSF2 3.552e+01 8.476e+00 4.191 3.05e-05 \*\*\*  
## BsmtUnfSF 2.770e+01 5.413e+00 5.117 3.79e-07 \*\*\*  
## TotalBsmtSF NA NA NA NA   
## X1stFlrSF 4.352e+01 6.244e+00 6.969 6.13e-12 \*\*\*  
## X2ndFlrSF 7.795e+01 7.238e+00 10.769 < 2e-16 \*\*\*  
## LowQualFinSF 1.560e+01 2.008e+01 0.777 0.437361   
## GrLivArea NA NA NA NA   
## BsmtFullBath 1.428e+03 2.377e+03 0.601 0.548044   
## BsmtHalfBath 5.026e+03 4.431e+03 1.134 0.256976   
## FullBath 5.593e+03 2.653e+03 2.108 0.035284 \*   
## HalfBath 3.348e+03 2.503e+03 1.337 0.181410   
## BedroomAbvGr -5.321e+03 1.654e+03 -3.217 0.001343 \*\*   
## KitchenAbvGr -8.862e+03 6.680e+03 -1.327 0.184948   
## TotRmsAbvGrd 4.537e+02 1.172e+03 0.387 0.698683   
## Fireplaces 5.303e+03 3.054e+03 1.736 0.082833 .   
## GarageCars 6.842e+03 2.614e+03 2.617 0.009006 \*\*   
## GarageArea 8.174e+00 8.926e+00 0.916 0.360016   
## WoodDeckSF 1.736e+01 7.013e+00 2.475 0.013492 \*   
## OpenPorchSF 2.910e+01 1.806e+01 1.611 0.107562   
## EnclosedPorch -1.696e+01 3.096e+01 -0.548 0.583933   
## X3SsnPorch 4.167e+01 2.493e+01 1.671 0.095019 .   
## ScreenPorch 6.477e+01 1.911e+01 3.389 0.000731 \*\*\*  
## PoolArea 4.396e+02 4.046e+02 1.086 0.277588   
## MiscVal 5.372e+00 6.195e+00 0.867 0.386056   
## YrSold -1.600e+03 6.106e+02 -2.621 0.008922 \*\*   
## NeighborhoodBlueste -3.091e+03 2.657e+04 -0.116 0.907431   
## NeighborhoodBrDale -1.352e+04 1.276e+04 -1.059 0.289750   
## NeighborhoodBrkSide -1.432e+04 1.073e+04 -1.334 0.182388   
## NeighborhoodClearCr -3.264e+04 1.113e+04 -2.933 0.003446 \*\*   
## NeighborhoodCollgCr -2.289e+04 9.287e+03 -2.465 0.013881 \*   
## NeighborhoodCrawfor 3.487e+03 1.037e+04 0.336 0.736859   
## NeighborhoodEdwards -2.741e+04 9.956e+03 -2.753 0.006028 \*\*   
## NeighborhoodGilbert -2.338e+04 9.584e+03 -2.439 0.014907 \*   
## NeighborhoodIDOTRR -2.545e+04 1.138e+04 -2.236 0.025572 \*   
## NeighborhoodMeadowV -7.937e+03 1.291e+04 -0.615 0.538873   
## NeighborhoodMitchel -3.478e+04 1.025e+04 -3.394 0.000719 \*\*\*  
## NeighborhoodNAmes -2.469e+04 9.676e+03 -2.552 0.010886 \*   
## NeighborhoodNoRidge 4.423e+03 1.052e+04 0.420 0.674275   
## NeighborhoodNPkVill 1.131e+02 1.327e+04 0.009 0.993204   
## NeighborhoodNridgHt -5.135e+03 9.591e+03 -0.535 0.592488   
## NeighborhoodNWAmes -3.205e+04 9.907e+03 -3.235 0.001259 \*\*   
## NeighborhoodOldTown -2.927e+04 1.032e+04 -2.836 0.004665 \*\*   
## NeighborhoodSawyer -2.463e+04 1.002e+04 -2.458 0.014166 \*   
## NeighborhoodSawyerW -2.171e+04 9.775e+03 -2.221 0.026569 \*   
## NeighborhoodSomerst -8.369e+03 9.663e+03 -0.866 0.386706   
## NeighborhoodStoneBr 2.303e+04 1.037e+04 2.220 0.026670 \*   
## NeighborhoodSWISU -2.542e+04 1.166e+04 -2.180 0.029503 \*   
## NeighborhoodTimber -2.396e+04 1.014e+04 -2.362 0.018365 \*   
## NeighborhoodVeenker 3.135e+03 1.311e+04 0.239 0.811056   
## ExterQualFa -1.282e+04 1.122e+04 -1.142 0.253602   
## ExterQualGd -2.072e+04 5.467e+03 -3.790 0.000161 \*\*\*  
## ExterQualTA -2.120e+04 6.106e+03 -3.472 0.000542 \*\*\*  
## BsmtQualFa -2.830e+04 7.341e+03 -3.855 0.000124 \*\*\*  
## BsmtQualGd -2.916e+04 4.240e+03 -6.878 1.13e-11 \*\*\*  
## BsmtQualNo Basement -6.905e+01 1.204e+04 -0.006 0.995426   
## BsmtQualTA -2.661e+04 5.170e+03 -5.147 3.25e-07 \*\*\*  
## KitchenQualFa -1.665e+04 7.587e+03 -2.194 0.028478 \*   
## KitchenQualGd -1.414e+04 4.111e+03 -3.438 0.000612 \*\*\*  
## KitchenQualTA -1.579e+04 4.634e+03 -3.406 0.000687 \*\*\*  
## GarageFinishNo Garage 2.728e+04 1.442e+04 1.892 0.058801 .   
## GarageFinishRFn -2.272e+03 2.451e+03 -0.927 0.354212   
## GarageFinishUnf -1.112e+02 2.974e+03 -0.037 0.970167   
## FireplaceQuFa -1.685e+04 8.197e+03 -2.055 0.040125 \*   
## FireplaceQuGd -1.055e+04 6.346e+03 -1.662 0.096812 .   
## FireplaceQuNo Fireplace -5.247e+03 7.699e+03 -0.681 0.495738   
## FireplaceQuPo -1.039e+04 9.108e+03 -1.141 0.254305   
## FireplaceQuTA -1.341e+04 6.606e+03 -2.031 0.042594 \*   
## FoundationCBlock 1.067e+03 3.749e+03 0.285 0.776090   
## FoundationPConc 3.567e+03 4.261e+03 0.837 0.402641   
## FoundationSlab -1.673e+03 1.140e+04 -0.147 0.883357   
## FoundationStone 1.630e+04 1.490e+04 1.094 0.274248   
## FoundationWood -2.517e+04 1.562e+04 -1.612 0.107413   
## GarageTypeAttchd 1.457e+04 1.321e+04 1.103 0.270326   
## GarageTypeBasment 2.484e+04 1.524e+04 1.629 0.103584   
## GarageTypeBuiltIn 2.037e+04 1.380e+04 1.476 0.140334   
## GarageTypeCarPort -4.575e+03 1.620e+04 -0.283 0.777612   
## GarageTypeDetchd 1.606e+04 1.319e+04 1.218 0.223547   
## GarageTypeNo Garage NA NA NA NA   
## MSSubClass30 3.000e+03 5.597e+03 0.536 0.592081   
## MSSubClass40 1.501e+04 2.559e+04 0.587 0.557652   
## MSSubClass45 -5.889e+02 9.821e+03 -0.060 0.952199   
## MSSubClass50 1.132e+04 6.850e+03 1.653 0.098775 .   
## MSSubClass60 1.552e+02 6.258e+03 0.025 0.980214   
## MSSubClass70 3.346e+03 7.943e+03 0.421 0.673630   
## MSSubClass75 1.201e+04 1.136e+04 1.057 0.290654   
## MSSubClass80 3.405e+03 4.860e+03 0.701 0.483685   
## MSSubClass85 3.022e+03 7.299e+03 0.414 0.678913   
## MSSubClass90 -1.141e+04 6.843e+03 -1.667 0.095879 .   
## MSSubClass120 -2.412e+04 4.814e+03 -5.010 6.56e-07 \*\*\*  
## MSSubClass160 -2.629e+04 8.470e+03 -3.104 0.001968 \*\*   
## MSSubClass180 -2.420e+04 1.250e+04 -1.936 0.053194 .   
## MSSubClass190 -3.008e+02 7.659e+03 -0.039 0.968679   
## BsmtFinType1BLQ 4.037e+03 3.244e+03 1.244 0.213695   
## BsmtFinType1GLQ 3.441e+03 3.023e+03 1.138 0.255308   
## BsmtFinType1LwQ -4.354e+03 4.460e+03 -0.976 0.329208   
## BsmtFinType1No Basement NA NA NA NA   
## BsmtFinType1Rec -3.443e+03 3.441e+03 -1.001 0.317269   
## BsmtFinType1Unf -2.760e+02 3.364e+03 -0.082 0.934644   
## univ\_outl\_count -3.571e+03 2.580e+03 -1.384 0.166733   
## f.LotArea.L 9.961e+03 3.883e+03 2.565 0.010474 \*   
## secondfloor1 -2.625e+04 7.258e+03 -3.617 0.000315 \*\*\*  
## OpenPorch\_binary1 1.191e+03 5.220e+03 0.228 0.819604   
## EnclosedPorch\_binary1 1.230e+04 7.320e+03 1.681 0.093087 .   
## HasPorch\_binary1 -3.971e+03 5.322e+03 -0.746 0.455718   
## Pool\_binary1 -2.599e+05 2.801e+05 -0.928 0.353823   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 23630 on 905 degrees of freedom  
## Multiple R-squared: 0.9192, Adjusted R-squared: 0.9094   
## F-statistic: 93.57 on 110 and 905 DF, p-value: < 2.2e-16

1. Process of adding categorical variables to the model

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

## [1] 0.9208199

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 12.703 26 2.5416 7.5028 < 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 -1484.587

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

## [1] 12.62944

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

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.975044 1 1.405362  
## log(LotArea) 3.181995 1 1.783815  
## OverallQual 3.644983 1 1.909184  
## OverallCond 1.345689 1 1.160038  
## YearBuilt 6.069206 1 2.463576  
## BsmtFinSF1 2.188352 1 1.479308  
## log(X1stFlrSF) 4.504226 1 2.122316  
## X2ndFlrSF 4.606928 1 2.146376  
## BsmtFullBath 1.903356 1 1.379622  
## FullBath 2.595355 1 1.611010  
## TotRmsAbvGrd 3.940294 1 1.985017  
## KitchenAbvGr 1.389762 1 1.178882  
## Fireplaces 1.681695 1 1.296802  
## GarageCars 2.139030 1 1.462542  
## ScreenPorch 1.091566 1 1.044780  
## PoolArea 1.040211 1 1.019907  
## Neighborhood 44.452433 24 1.082259

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] 0.909216

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 14.565 5 1.6316 22.315 < 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 -1387.641

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

## [1] 11.01516

vif(b1)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.876880 1 1.369993  
## log(LotArea) 2.090942 1 1.446009  
## OverallQual 3.558948 1 1.886517  
## OverallCond 1.289359 1 1.135500  
## YearBuilt 2.580533 1 1.606404  
## BsmtFinSF1 2.064861 1 1.436963  
## log(X1stFlrSF) 4.030907 1 2.007712  
## X2ndFlrSF 4.043064 1 2.010737  
## BsmtFullBath 1.829917 1 1.352744  
## FullBath 2.351956 1 1.533609  
## TotRmsAbvGrd 3.740286 1 1.933982  
## KitchenAbvGr 1.317854 1 1.147978  
## Fireplaces 1.486765 1 1.219330  
## GarageCars 1.994584 1 1.412297  
## ScreenPorch 1.061171 1 1.030131  
## PoolArea 1.021649 1 1.010766  
## ExterQual 2.939246 3 1.196849

# 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] 0.9165131

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 14.565   
## 2 958 13.394 38 1.1707 2.2035 4.844e-05 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b1,b11)

## df AIC  
## b1 21 -1387.641  
## b11 59 -1396.775

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] 0.9138978

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 14.565   
## 2 992 13.814 4 0.75112 13.485 1.045e-10 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b1,b2)

## df AIC  
## b1 21 -1387.641  
## b2 25 -1433.435

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

## [1] 11.6141

vif(b2)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.891514 1 1.375323  
## log(LotArea) 2.108226 1 1.451973  
## OverallQual 3.826560 1 1.956159  
## OverallCond 1.310590 1 1.144810  
## YearBuilt 3.302730 1 1.817341  
## BsmtFinSF1 2.118895 1 1.455642  
## log(X1stFlrSF) 4.036522 1 2.009110  
## X2ndFlrSF 4.115220 1 2.028600  
## BsmtFullBath 1.844596 1 1.358159  
## FullBath 2.462512 1 1.569239  
## TotRmsAbvGrd 3.889954 1 1.972297  
## KitchenAbvGr 1.418151 1 1.190861  
## Fireplaces 1.491174 1 1.221137  
## GarageCars 2.025469 1 1.423190  
## ScreenPorch 1.064332 1 1.031665  
## PoolArea 1.021988 1 1.010934  
## ExterQual 3.820575 3 1.250321  
## BsmtQual 5.710185 4 1.243315

# 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] 0.9209316

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 13.814   
## 2 929 12.685 63 1.1285 1.3118 0.0564 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b2,b21)

## df AIC  
## b2 25 -1433.435  
## b21 88 -1394.021

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 13.394   
## 2 929 12.685 29 0.70889 1.7902 0.00661 \*\*  
## ---  
## 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] 0.9151036

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 13.814   
## 2 989 13.620 3 0.19347 4.6826 0.002966 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b2,b3)

## df AIC  
## b2 25 -1433.435  
## b3 28 -1441.765

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

## [1] 11.77907

vif(b3)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.910426 1 1.382182  
## log(LotArea) 2.114588 1 1.454162  
## OverallQual 3.910296 1 1.977447  
## OverallCond 1.379947 1 1.174712  
## YearBuilt 3.384395 1 1.839672  
## BsmtFinSF1 2.127727 1 1.458673  
## log(X1stFlrSF) 4.084895 1 2.021112  
## X2ndFlrSF 4.151761 1 2.037587  
## BsmtFullBath 1.852804 1 1.361177  
## FullBath 2.498905 1 1.580792  
## TotRmsAbvGrd 3.913272 1 1.978199  
## KitchenAbvGr 1.437957 1 1.199148  
## Fireplaces 1.492497 1 1.221678  
## GarageCars 2.068284 1 1.438153  
## ScreenPorch 1.067635 1 1.033264  
## PoolArea 1.024958 1 1.012402  
## ExterQual 5.799606 3 1.340396  
## BsmtQual 6.308179 4 1.258891  
## KitchenQual 4.897612 3 1.303159

#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] 0.9240053

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 13.620   
## 2 931 12.192 58 1.4282 1.8802 0.0001171 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b3,b31)

## df AIC  
## b3 28 -1441.765  
## b31 86 -1438.305

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] 0.9153875

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 13.620   
## 2 986 13.575 3 0.045544 1.1027 0.3471

AIC(b3,b4)

## df AIC  
## b3 28 -1441.765  
## b4 31 -1439.168

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

## [1] 11.81859

vif(b4)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.939674 1 1.392722  
## log(LotArea) 2.116689 1 1.454885  
## OverallQual 3.923870 1 1.980876  
## OverallCond 1.383818 1 1.176358  
## YearBuilt 3.611757 1 1.900462  
## BsmtFinSF1 2.138219 1 1.462265  
## log(X1stFlrSF) 4.099619 1 2.024752  
## X2ndFlrSF 4.160374 1 2.039699  
## BsmtFullBath 1.858173 1 1.363148  
## FullBath 2.567372 1 1.602302  
## TotRmsAbvGrd 3.948648 1 1.987121  
## KitchenAbvGr 1.459205 1 1.207976  
## Fireplaces 1.521064 1 1.233314  
## GarageCars 3.045998 1 1.745279  
## ScreenPorch 1.069844 1 1.034333  
## PoolArea 1.026164 1 1.012998  
## ExterQual 6.006461 3 1.348248  
## BsmtQual 6.620103 4 1.266508  
## KitchenQual 4.987321 3 1.307107  
## GarageFinish 3.843917 3 1.251591

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] 0.9159415

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 13.620   
## 2 984 13.486 5 0.13442 1.9616 0.08189 .  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b3,b5)

## df AIC  
## b3 28 -1441.765  
## b5 33 -1441.842

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

## [1] 11.89647

vif(b5)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.922075 1 1.386389  
## log(LotArea) 2.130818 1 1.459732  
## OverallQual 3.967632 1 1.991892  
## OverallCond 1.383209 1 1.176099  
## YearBuilt 3.447975 1 1.856872  
## BsmtFinSF1 2.159877 1 1.469652  
## log(X1stFlrSF) 4.130147 1 2.032276  
## X2ndFlrSF 4.244683 1 2.060263  
## BsmtFullBath 1.864435 1 1.365443  
## FullBath 2.517593 1 1.586692  
## TotRmsAbvGrd 3.972748 1 1.993175  
## KitchenAbvGr 1.453705 1 1.205697  
## Fireplaces 5.965429 1 2.442423  
## GarageCars 2.074162 1 1.440195  
## ScreenPorch 1.077702 1 1.038124  
## PoolArea 1.043628 1 1.021581  
## ExterQual 5.884719 3 1.343655  
## BsmtQual 6.684523 4 1.268042  
## KitchenQual 5.088884 3 1.311506  
## FireplaceQu 8.901560 5 1.244362

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] 0.9166424

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 13.620   
## 2 984 13.374 5 0.24688 3.6329 0.002914 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b3,b6)

## df AIC  
## b3 28 -1441.765  
## b6 33 -1450.349

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

## [1] 11.99651

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

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.964707 1 1.401680  
## log(LotArea) 2.123062 1 1.457073  
## OverallQual 4.026878 1 2.006708  
## OverallCond 1.433789 1 1.197409  
## YearBuilt 5.499525 1 2.345107  
## BsmtFinSF1 2.162105 1 1.470410  
## log(X1stFlrSF) 4.153418 1 2.037994  
## X2ndFlrSF 4.214722 1 2.052979  
## BsmtFullBath 1.857537 1 1.362915  
## FullBath 2.534110 1 1.591889  
## TotRmsAbvGrd 3.929935 1 1.982406  
## KitchenAbvGr 1.439979 1 1.199991  
## Fireplaces 1.524141 1 1.234561  
## GarageCars 2.114880 1 1.454263  
## ScreenPorch 1.071616 1 1.035189  
## PoolArea 1.028517 1 1.014158  
## ExterQual 6.082466 3 1.351077  
## BsmtQual 19.311311 4 1.447860  
## KitchenQual 5.180014 3 1.315392  
## Foundation 17.211572 5 1.329175

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] 0.9162176

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 13.620   
## 2 983 13.442 6 0.17872 2.1784 0.04287 \*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b3,b7)

## df AIC  
## b3 28 -1441.765  
## b7 34 -1443.185

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

## [1] 11.93568

vif(b7)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.973582 1 1.404842  
## log(LotArea) 2.145056 1 1.464601  
## OverallQual 3.959355 1 1.989813  
## OverallCond 1.391345 1 1.179553  
## YearBuilt 3.981528 1 1.995377  
## BsmtFinSF1 2.144812 1 1.464518  
## log(X1stFlrSF) 4.190035 1 2.046958  
## X2ndFlrSF 4.365937 1 2.089483  
## BsmtFullBath 1.857983 1 1.363078  
## FullBath 2.542579 1 1.594547  
## TotRmsAbvGrd 3.952111 1 1.987992  
## KitchenAbvGr 1.505556 1 1.227011  
## Fireplaces 1.525563 1 1.235137  
## GarageCars 3.235075 1 1.798632  
## ScreenPorch 1.073448 1 1.036074  
## PoolArea 1.025807 1 1.012821  
## ExterQual 6.017820 3 1.348673  
## BsmtQual 6.681531 4 1.267972  
## KitchenQual 5.104183 3 1.312163  
## GarageType 4.873570 6 1.141092

# 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] 0.9271821

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 13.442   
## 2 919 11.683 64 1.7591 2.1621 9.004e-07 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

AIC(b7,b71)

## df AIC  
## b7 34 -1443.185  
## b71 98 -1457.690

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

## [1] 13.73289

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] 0.9177258

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 13.442   
## 2 969 13.200 14 0.24197 1.2688 0.2203

AIC(b7,b8)

## df AIC  
## b7 34 -1443.185  
## b8 48 -1433.641

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

## [1] 12.15448

vif(b8) # highly correlated

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 2.080840 1 1.442512  
## log(LotArea) 3.016287 1 1.736746  
## OverallQual 4.201448 1 2.049743  
## OverallCond 1.450824 1 1.204502  
## YearBuilt 8.544122 1 2.923033  
## BsmtFinSF1 2.189063 1 1.479548  
## log(X1stFlrSF) 4.847325 1 2.201664  
## X2ndFlrSF 9.272223 1 3.045033  
## BsmtFullBath 1.930861 1 1.389554  
## FullBath 2.600309 1 1.612547  
## TotRmsAbvGrd 4.208325 1 2.051420  
## KitchenAbvGr 2.962741 1 1.721262  
## Fireplaces 1.604690 1 1.266764  
## GarageCars 3.313643 1 1.820341  
## ScreenPorch 1.083640 1 1.040980  
## PoolArea 1.043489 1 1.021513  
## ExterQual 6.399306 3 1.362560  
## BsmtQual 8.329841 4 1.303406  
## KitchenQual 5.460015 3 1.326984  
## GarageType 7.032323 6 1.176499  
## MSSubClass 402.171157 14 1.238838

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] 0.9166285

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 13.442   
## 2 978 13.376 5 0.065924 0.964 0.4388

AIC(b7,b9)

## df AIC  
## b7 34 -1443.185  
## b9 39 -1438.180

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

## [1] 11.99451

#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] 0.9162254

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 13.442   
## 2 982 13.441 1 0.0012519 0.0915 0.7624

AIC(b7,b10)

## df AIC  
## b7 34 -1443.185  
## b10 35 -1441.280

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

## [1] 11.93679

vif(b10)

## GVIF Df GVIF^(1/(2\*Df))  
## LotFrontage 1.973758 1 1.404905  
## log(LotArea) 2.470702 1 1.571847  
## OverallQual 3.976800 1 1.994191  
## OverallCond 1.391553 1 1.179641  
## YearBuilt 3.982592 1 1.995643  
## BsmtFinSF1 2.153192 1 1.467376  
## log(X1stFlrSF) 4.194624 1 2.048078  
## X2ndFlrSF 4.391481 1 2.095586  
## BsmtFullBath 1.862506 1 1.364737  
## FullBath 2.558484 1 1.599526  
## TotRmsAbvGrd 3.987432 1 1.996856  
## KitchenAbvGr 1.506087 1 1.227227  
## Fireplaces 1.532962 1 1.238128  
## GarageCars 3.235752 1 1.798820  
## ScreenPorch 1.073478 1 1.036088  
## PoolArea 1.036151 1 1.017915  
## ExterQual 6.033805 3 1.349269  
## BsmtQual 6.694410 4 1.268277  
## KitchenQual 5.128524 3 1.313204  
## GarageType 4.904200 6 1.141688  
## f.LotArea 1.381049 1 1.175181

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] 0.9165131

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 13.442   
## 2 958 13.394 25 0.04741 0.1356 1

AIC(b7,b11)

## df AIC  
## b7 34 -1443.185  
## b11 59 -1396.775

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

## df AIC  
## b7 34 -1443.185  
## b71 98 -1457.690  
## b3 28 -1441.765  
## b31 86 -1438.305  
## b2 25 -1433.435  
## b21 88 -1394.021  
## b1 21 -1387.641  
## b11 59 -1396.775

final\_model<-b3

1. Functions

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)  
}

1. Test preprocess

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")

imputed\_test<-complete(mice\_imp)