House Prices

SIM - Assignment 1

Javier Herrer Torres, Marc Fortó Cornella, Max Ticó Miñarro

2023-11-19

Contents

1 Data preparation **1**
1.1 Variable Analysis . . . 4

2 Data quality report **22**
2.1 Missing Imputation . . . 22
2.2 Observed relations . . . 22
2.3 Univariate Outliers . . . 22
2.4 Multivariate Outliers . . . 23

3 Profiling **24**
3.1 Determine if the response variable (price) has an acceptably normal distribution. Address test to discard serial correlation. 24
3.2 Categorize numeric variables. . . . 25
3.3 Interactions between categorical and numerical variables . . . 26

4 Price Modelling **27**
4.1 Model building . . . 27
4.2 Multicollinearity on the model . . . 27

5 Model validation **30**

6 Residual outliers **32**

7 A priori influential data observations **34**

8 A posteriori influential data observations **34**

9 Model testing with test samples **36**
9.1 Load and prepare Test Data . . . 36
9.2 Make predictions . . . 40

```
rm(list = ls())
house_prices <- read.csv("train.csv")
par(mfrow = c(1, 1))
```

1 Data preparation

First, the training data was imported through the `read.csv` function.

Then, 10 factors are selected using the continuos description method and filtering by the 10 most related factors to the target. Before that, factors should have the appropiate type. The selected factors are:

1. overall material and finish of the house,
2. physical locations within the Ames city limits,
3. quality of the material on the exterior,
4. basement height evaluation,
5. kitchen quality,
6. interior finish of the garage,
7. fireplace quality,
8. type of foundation,
9. garage location,
10. type of dwelling involved in the sale.

```
library(tidyr)
```

```r
na_factor_cols <- c("BsmtQual", "GarageFinish", "FireplaceQu", "GarageType")

house_prices[na_factor_cols] <- lapply(
  house_prices[na_factor_cols],
  function(x) {
    replace_na(x, "NA")
  }
)

house_prices$MSSubClass <- factor(house_prices$MSSubClass)
house_prices$MSZoning <- factor(house_prices$MSZoning)
house_prices$Street <- factor(house_prices$Street)
house_prices$Alley <- factor(house_prices$Alley)
house_prices$LotShape <- factor(house_prices$LotShape)
house_prices$LandContour <- factor(house_prices$LandContour)
house_prices$Utilities <- factor(house_prices$Utilities)
house_prices$LotConfig <- factor(house_prices$LotConfig)
house_prices$LandSlope <- factor(house_prices$LandSlop)
house_prices$Neighborhood <- factor(house_prices$Neighborhood)
house_prices$Condition1 <- factor(house_prices$Condition1)
house_prices$Condition2 <- factor(house_prices$Condition2)
house_prices$BldgType <- factor(house_prices$BldgType)
house_prices$HouseStyle <- factor(house_prices$HouseStyle)
house_prices$OverallQual <- factor(house_prices$OverallQual)
house_prices$OverallCond <- factor(house_prices$OverallCond)
house_prices$RoofStyle <- factor(house_prices$RoofStyle)
house_prices$RoofMatl <- factor(house_prices$RoofMatl)
house_prices$Exterior1st <- factor(house_prices$Exterior1st)
house_prices$Exterior2nd <- factor(house_prices$Exterior2nd)
house_prices$MasVnrType <- factor(house_prices$MasVnrType)
house_prices$ExterQual <- factor(house_prices$ExterQual)
house_prices$ExterCond <- factor(house_prices$ExterCond)
house_prices$Foundation <- factor(house_prices$Foundation)
house_prices$BsmtCond <- factor(house_prices$BsmtCond)
house_prices$BsmtExposure <- factor(house_prices$BsmtExposure)
house_prices$BsmtFinType1 <- factor(house_prices$BsmtFinType1)
house_prices$BsmtFinType2 <- factor(house_prices$BsmtFinType2)
house_prices$Heating <- factor(house_prices$Heating)
house_prices$HeatingQC <- factor(house_prices$HeatingQC)
house_prices$CentralAir <- factor(house_prices$CentralAir)
house_prices$Electrical <- factor(house_prices$Electrical)
house_prices$KitchenQual <- factor(house_prices$KitchenQual)
house_prices$Functional <- factor(house_prices$Functional)
house_prices$FireplaceQu <- factor(house_prices$FireplaceQu)
house_prices$GarageFinish <- factor(house_prices$GarageFinish)
house_prices$GarageQual <- factor(house_prices$GarageQual)
house_prices$Heating <- factor(house_prices$Heating)
house_prices$GarageCond <- factor(house_prices$GarageCond)
house_prices$PavedDrive <- factor(house_prices$PavedDrive)
house_prices$PoolQC <- factor(house_prices$PoolQC)
house_prices$Fence <- factor(house_prices$Fence)
house_prices$MiscFeature <- factor(house_prices$MiscFeature)
house_prices$SaleType <- factor(house_prices$SaleType)
house_prices$SaleCondition <- factor(house_prices$SaleCondition)

continuos_description <- condes(house_prices, 81)
# continuos_description$quali

relevant_factors <- rownames(continuos_description$quali[1:10, ])
relevant_factors
```

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

```r
numeric_variables <- sapply(house_prices, is.numeric)

house_prices <- cbind(
  house_prices[, numeric_variables],
```

```r
  house_prices[, relevant_factors]
)
```

Now, we add the levels for the selected factors.

```r
cols <- c(
  "OverallQual", "Neighborhood", "ExterQual", "BsmtQual", "KitchenQual",
  "GarageFinish", "FireplaceQu", "Foundation", "GarageType", "MSSubClass"
)

levels_list <- list(
  1:10, # OverallQual
  c(
    "Blmngtn", "Blueste", "BrDale", "BrkSide", "ClearCr", "CollgCr", "Crawfor",
    "Edwards", "Gilbert", "IDOTRR", "MeadowV", "Mitchel", "NAmes", "NoRidge",
    "NPkVill", "NridgHt", "NWAmes", "OldTown", "SWISU", "Sawyer", "SawyerW",
    "Somerst", "StoneBr", "Timber", "Veenker"
  ), # Neighborhood
  c("Ex", "Gd", "TA", "Fa", "Po"), # ExterQual
  c("Ex", "Gd", "TA", "Fa", "Po", "NA"), # BsmtQual
  c("Ex", "Gd", "TA", "Fa", "Po"), # KitchenQual
  c("Fin", "RFn", "Unf", "NA"), # GarageFinish
  c("Ex", "Gd", "TA", "Fa", "Po", "NA"), # FireplaceQu
  c("BrkTil", "CBlock", "PConc", "Slab", "Stone", "Wood"), # Foundation
  c(
    "2Types", "Attchd", "Basment", "BuiltIn", "CarPort", "Detchd", "NA"
  ), # GarageType
  c(
    "20", "30", "40", "45", "50", "60", "70", "75", "80", "85", "90", "120",
    "150", "160", "180", "190"
  ) # MSSubClass
)

labels_list <- list(
  c(
    "Very Poor", "Poor", "Fair", "Below Average", "Average", "Above Average",
    "Good", "Very Good", "Excellent", "Very Excellent"
  ), # OverallQual
  c(
    "Bloomington Heights", "Bluestem", "Briardale", "Brookside", "Clear Creek",
    "College Creek", "Crawford", "Edwards", "Gilbert", "Iowa DOT and Rail Road",
    "Meadow Village", "Mitchell", "North Ames", "Northridge", "Northpark Villa",
    "Northridge Heights", "Northwest Ames", "Old Town",
    "South & West of Iowa State University", "Sawyer", "Sawyer West",
    "Somerset", "Stone Brook", "Timberland", "Veenker"
  ), # Neighborhood
  c("Excellent", "Good", "Average/Typical", "Fair", "Poor"), # ExterQual
  c(
    "Excellent (100+ inches)", "Good (90-99 inches)", "Typical (80-89 inches)",
    "Fair (70-79 inches)", "Poor (<70 inches)", "No Basement"
  ), # BsmtQual
  c("Excellent", "Good", "Typical/Average", "Fair", "Poor"), # KitchenQual
  c("Finished", "Rough Finished", "Unfinished", "No Garage"), # GarageFinish
  c(
    "Excellent",
    "Good",
    "Average", # nolint: line_length_linter.
    "Fair",
    "Poor",
    "No Fireplace"
  ), # FireplaceQu
  c(
    "Brick & Tile", "Cinder Block", "Poured Contrete", "Slab", "Stone", "Wood"
  ), # Foundation
  c(
    "More than one type of garage", "Attached to home", "Basement Garage",
    "Built-In (Garage part of house - typically has room above garage)",
    "Car Port", "Detached from home", "No Garage"
```

```
  ), # GarageType
  c(
    "1-STORY 1946 & NEWER ALL STYLES", "1-STORY 1945 & OLDER",
    "1-STORY W/FINISHED ATTIC ALL AGES", "1-1/2 STORY - UNFINISHED ALL AGES",
    "1-1/2 STORY FINISHED ALL AGES", "2-STORY 1946 & NEWER",
    "2-STORY 1945 & OLDER", "2-1/2 STORY ALL AGES", "SPLIT OR MULTI-LEVEL",
    "SPLIT FOYER",
    "DUPLEX - ALL STYLES AND AGES",
    "1-STORY PUD (Planned Unit Development) - 1946 & NEWER",
    "1-1/2 STORY PUD - ALL AGES", "2-STORY PUD - 1946 & NEWER",
    "PUD - MULTILEVEL - INCL SPLIT LEV/FOYER",
    "2 FAMILY CONVERSION - ALL STYLES AND AGES"
  ) # MSSubClass
)

house_prices[cols] <- lapply(
  seq_along(cols),
  function(i) {
    factor(
      house_prices[[cols[i]]],
      levels = levels_list[[i]],
      labels = labels_list[[i]]
    )
  }
)
```

1.1 Variable Analysis

variable 1: LotFrontage

LotFrontage is a numerical variable with 259 NA's. Then we used a histogram and a Boxplot to visualize the distribution of the values of this variable. By using a Shapiro test we observed a non-normal distribution for LotFrontage (p-value near 0). Afterwards, we computed the InterQuartileRange to build the thresholds for severe outliers. 88 outliers were observed, from which 12 were severe outliers.

```
summary(house_prices$LotFrontage)
```

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

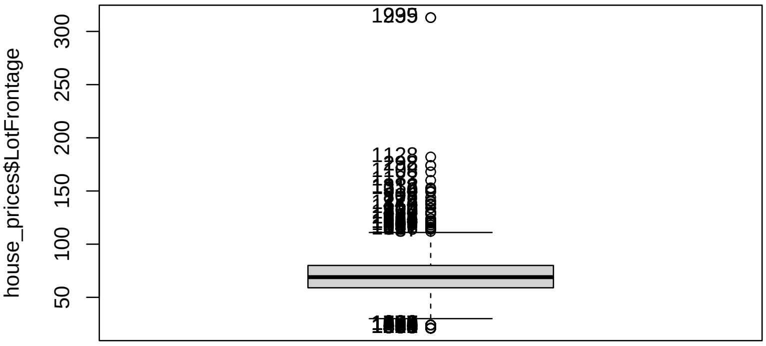
```
# Histogram plotting
# hist(house_prices$LotFrontage,
#   main = "Linear feet of street connected to property",
#   xlab = "Number of feet",
#   ylab = "Frequency"
# )

# Missing values
sum(is.na(house_prices$LotFrontage))
```

```
## [1] 259
```

```
# Checking for normal distribution
shapiro.test(house_prices$LotFrontage)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$LotFrontage
## W = 0.8804, p-value < 2.2e-16
```

```
# Univariant Outliers
length(Boxplot(house_prices$LotFrontage, id = list(n = Inf)))
```




```
## [1] 88
```

```r
varout <- summary(house_prices$LotFrontage)
iqr <- varout[5] - varout[2]
sev_up <- varout[5] + 3 * iqr
sev_down <- varout[2] - 3 * iqr

# Number of severe outliers
length(which(house_prices$LotFrontage > sev_up)) + length(which(house_prices$LotFrontage < sev_down))
```

```
## [1] 12
```

variable 2: LotArea

LotArea is a numerical variable with 0 NA's. Then we used a histogram and a Boxplot to visualize the distribution of the values of this variable. By using a Shapiro test we observed a non-normal distribution for LotArea (p-value < 2.2e-16). Afterwards, we computed the InterQuartileRange to build the thresholds for severe outliers. 68 outliers were observed,from which 34 were severe outliers.

```r
summary(house_prices$LotArea)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1300    7554    9478   10517   11602  215245
```

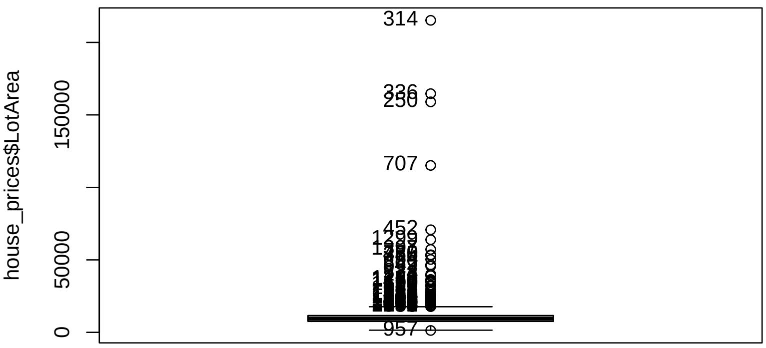
```r
# Histogram plotting
# hist(house_prices$LotArea,
#   main = "Lot size in square feet",
#   xlab = "Number of feet",
#   ylab = "Density",
#   freq = F
# )

# Missing values
sum(is.na(house_prices$LotArea))
```

```
## [1] 0
```

```r
# Checking for normal distribution
shapiro.test(house_prices$LotArea)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$LotArea
## W = 0.35106, p-value < 2.2e-16
```

```r
# Univariant Outliers
length(Boxplot(house_prices$LotArea, id = list(n = Inf)))
```




```
## [1] 68
```

```r
# Boxplot(house_prices$LotArea, id = list(n = Inf))
sev_up <- (quantile(house_prices$LotArea, 0.75) + (3 * ((quantile(house_prices$LotArea, 0.75) - quantile(house_p
sev_down <- (quantile(house_prices$LotArea, 0.25) - (3 * ((quantile(house_prices$LotArea, 0.75) - quantile(house
length(which(house_prices$LotArea > sev_up))
```

```
## [1] 34
```

```r
length(which(house_prices$LotArea < sev_down))
```

```
## [1] 0
```

```r
ll <- house_prices[which(house_prices$LotArea > sev_up), ]
```

variable 3: YearBuilt

YearBuilt is a numeric interval variable. By using a Shapiro test we observed a non-normal distribution for YearBuilt (p-value < 2.2e-16). Afterwards, we computed the InterQuartileRange to build the thresholds for severe outliers. 7 outliers were observed,from which 0 were severe outliers.

```r
summary(house_prices$YearBuilt)
```

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

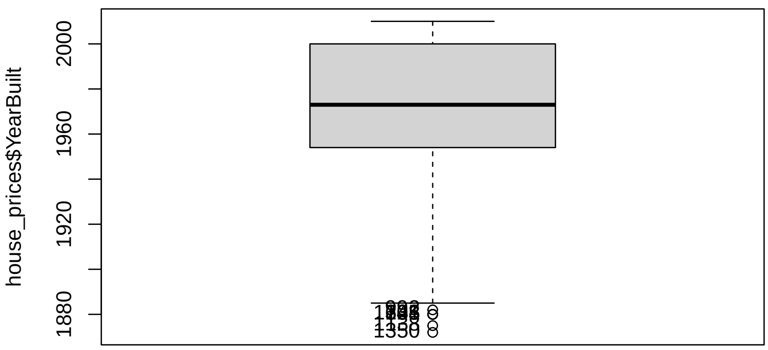
```r
# Histogram plotting
# hist(house_prices$YearBuilt,
#   main = "year of construction",
#   xlab = "Year",
#   ylab = "Density",
#   freq = F
# )
# curve(dnorm(x, mean(house_prices$YearBuilt), sd(house_prices$YearBuilt)), add = TRUE, col = "red")

# Missing values
sum(is.na(house_prices$YearBuilt))
```

```
## [1] 0
```

```r
# Checking for normal distribution
shapiro.test(house_prices$YearBuilt)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$YearBuilt
## W = 0.9256, p-value < 2.2e-16
```

```r
# Univariant Outliers
length(Boxplot(house_prices$YearBuilt, id = list(n = Inf)))
```




```
## [1] 7
```

```r
# Boxplot(house_prices$YearBuilt, id = list(n = Inf))
sev_down <- (quantile(house_prices$YearBuilt, 0.25) - (3 * ((quantile(house_prices$YearBuilt, 0.75) - quantile(h
length(which(house_prices$YearBuilt < sev_down))
```

```
## [1] 0
```

variable 4: YearRemodAdd

YearRemodAdd is a numeric interval variable with 0 NA's. By using a Shapiro test we observed a non-normal distribution for YearRemodAdd (p-value < 2.2e-16). We did not observe any outlier for this variable.

```r
summary(house_prices$YearRemodAdd)
```

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

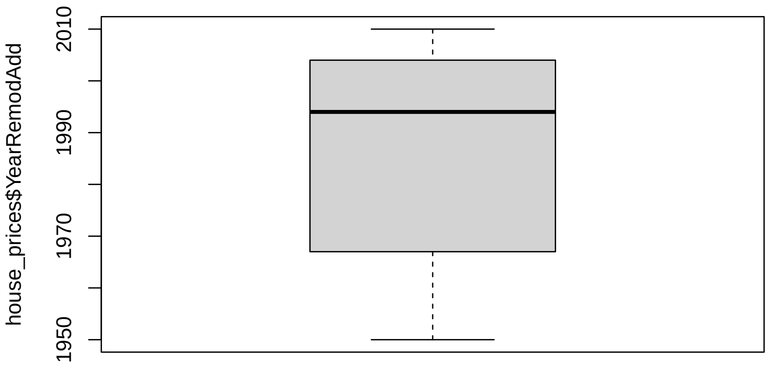
```r
# Histogram plotting
# hist(house_prices$YearRemodAdd,
#   main = "Remodel year",
#   xlab = "Year",
#   ylab = "Density",
#   freq = F
# )
# curve(dnorm(x, mean(house_prices$YearRemodAdd), sd(house_prices$YearRemodAdd)), add = TRUE, col = "red")

# Missing values
sum(is.na(house_prices$YearRemodAdd))
```

```
## [1] 0
```

```r
# Checking for normal distribution
shapiro.test(house_prices$YearRemodAdd)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$YearRemodAdd
## W = 0.8628, p-value < 2.2e-16
```

```r
# Univariant Outliers
length(Boxplot(house_prices$YearRemodAdd, id = list(n = Inf)))
```




```
## [1] 0
```

```r
# Boxplot(house_prices$YearRemodAdd, id = list(n = Inf))
```

variable 5: MasVnrArea

MasVnrArea is a numerical variable with 8 NA's. Then we used a histogram and a Boxplot to visualize the distribution of the values of this variable. By using a Shapiro test we observed a non-normal distribution for MasVnrArea (p-value < 2.2e-16). Afterwards, we computed the InterQuartileRange to build the thresholds for severe outliers. 68 outliers were observed,from which 34 were severe outliers.

```r
summary(house_prices$MasVnrArea)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.0     0.0     0.0   103.7   166.0  1600.0       8
```

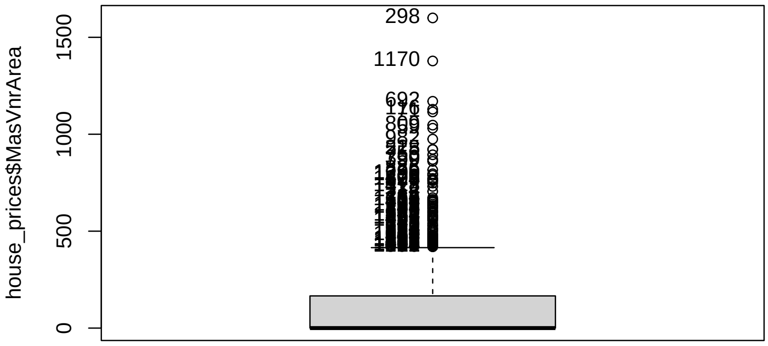
```r
# Histogram plotting
# hist(house_prices$MasVnrArea,
#   main = "Masonry veneer area in square feet",
#   xlab = "Square feet",
#   ylab = "Density",
#   freq = T
# )

# Missing values
sum(is.na(house_prices$MasVnrArea))
```

```
## [1] 8
```

```r
# Checking for normal distribution
shapiro.test(house_prices$MasVnrArea)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$MasVnrArea
## W = 0.63929, p-value < 2.2e-16
```

```r
# Univariant Outliers
length(Boxplot(house_prices$MasVnrArea, id = list(n = Inf)))
```




```
## [1] 96
```

```r
# Boxplot(house_prices$MasVnrArea, id = list(n = Inf))
varout <- summary(house_prices$MasVnrArea)
iqr <- varout[5] - varout[2]
sev_up <- varout[5] + 3 * iqr
sev_down <- varout[2] - 3 * iqr

# Number of severe outliers
length(which(house_prices$MasVnrArea > sev_up)) + length(which(house_prices$MasVnrArea < sev_down))
```

```
## [1] 25
```

variable 6: BsmtFinSF1

BsmtFinSF1 is a numerical variable. We observed that some values contained 0 values, but we decided not to declare them as NA, because they corresponded to BsmtFinSF2. In total, we had no NA's. Then we used a histogram and a Boxplot to visualize the distribution of the values of this variable. By using a Shapiro test we observed a non-normal distribution for BsmtFinSF1 (p-value < 2.2e-16). Afterwards, we computed the InterQuartileRange to build the thresholds for severe outliers. 13 outliers were observed, from which only 1 was a severe outlier.

```r
summary(house_prices$BsmtFinSF1)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0   383.5   443.6   712.2  5644.0
```

```r
# Histogram plotting
# hist(house_prices$BsmtFinSF1,
#   main = "Type 1 finished_square_feet",
#   xlab = "Square feet",
#   ylab = "Density",
#   freq = F
# )
# curve(dnorm(x, mean(house_prices$BsmtFinSF1), sd(house_prices$BsmtFinSF1)), add = TRUE, col = "red")

# Missing values
sum(is.na(house_prices$BsmtFinSF1))
```

```
## [1] 0
```

```r
# Checking for normal distribution
shapiro.test(house_prices$BsmtFinSF1)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$BsmtFinSF1
## W = 0.84796, p-value < 2.2e-16
```

```r
# Univariant Outliers
# length(Boxplot(house_prices$BsmtFinSF1, id = list(n = Inf)))
varout <- summary(house_prices$BsmtFinSF1)
iqr <- varout[5] - varout[2]
sev_up <- varout[5] + 3 * iqr
sev_down <- varout[2] - 3 * iqr

# Number of severe outliers
length(which(house_prices$BsmtFinSF1 > sev_up)) + length(which(house_prices$BsmtFinSF1 < sev_down))
```

```
## [1] 1
```

variable 7: BsmtFinSF2

BsmtFinSF2 is a numerical variable. We observed that some values contained 0 values, so we declared them as missing data. In total, we had 467 NA's. Then we used a histogram and a Boxplot to visualize the distribution of the values of this variable. By using a Shapiro test we observed a non-normal distribution for BsmtFinSF1 (p-value < 2.2e-16). We observed so many outliers (167), but they corresponded to those rows which had BsmtFinSF1.

```r
summary(house_prices$BsmtFinSF2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00   46.55    0.00 1474.00
```

```r
# Histogram plotting
# hist(house_prices$BsmtFinSF2,
#   main = "Type 2 finished_square_feet",
#   xlab = "Square feet",
#   ylab = "Density",
#   freq = F
# )
# curve(dnorm(x, mean(house_prices$BsmtFinSF2), sd(house_prices$BsmtFinSF2)), add = TRUE, col = "red")

# Missing values
sum(is.na(house_prices$BsmtFinSF2))
```

```
## [1] 0
```

```r
# Checking for normal distribution
shapiro.test(house_prices$BsmtFinSF2)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$BsmtFinSF2
## W = 0.32728, p-value < 2.2e-16
```

```r
# Univariant Outliers
# length(Boxplot(house_prices$BsmtFinSF2, id = list(n = Inf)))
varout <- summary(house_prices$BsmtFinSF2)
iqr <- varout[5] - varout[2]
sev_up <- varout[5] + 3 * iqr
sev_down <- varout[2] - 3 * iqr

# Number of severe outliers
length(which(house_prices$BsmtFinSF2 > sev_up)) + length(which(house_prices$BsmtFinSF2 < sev_down))
```

```
## [1] 167
```

variable 8: BsmtUnfSF

BsmtUnfSF is a numerical variable with 0 NA's. Then we used a histogram and a Boxplot to visualize the distribution of the values of this variable. By using a Shapiro test we observed a non-normal distribution for BsmtUnfSF (p-value < 2.2e-16). Afterwards, we computed the InterQuartileRange to build the thresholds for severe outliers. 29 outliers were observed,from which none of them were severe outliers.

```r
summary(house_prices$BsmtUnfSF)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   223.0   477.5   567.2   808.0  2336.0
```

```r
# Missing values
sum(is.na(house_prices$BsmtUnfSF))
```

```
## [1] 0
```

```r
# Checking for normal distribution
shapiro.test(house_prices$BsmtUnfSF)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$BsmtUnfSF
## W = 0.93042, p-value < 2.2e-16
```

variable 9: TotalBsmtSF

TotalBsmtSF is a numerical variable with no missing values. We first verified the coherence between the other Basement area information variables. Then we used a histogram and a Boxplot to visualize the distribution of the values of this variable. By using a Shapiro test we observed a non-normal distribution for TotalBsmtSF (p-value < 2.2e-16). Afterwards, we computed the InterQuartileRange to build the thresholds for severe outliers. 61 outliers were observed,from which 5 of them were severe outliers.

```r
summary(house_prices$TotalBsmtSF)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   795.8   991.5  1057.4  1298.2  6110.0
```

```r
ll <- which(house_prices$BsmtFinSF1 + house_prices$BsmtFinSF2 + house_prices$BsmtUnfSF != house_prices$TotalBsmt
ll
```

```
## integer(0)
```

```r
# Missing values
sum(is.na(house_prices$TotalBsmtSF))
```

```
## [1] 0
```

```r
# Checking for normal distribution
shapiro.test(house_prices$TotalBsmtSF)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$TotalBsmtSF
## W = 0.91735, p-value < 2.2e-16
```

variable 10: X1stFlrSF

X1stFlrSF is a numerical variable with no missing values. Then we used a histogram and a Boxplot to visualize the distribution of the values of this variable. By using a Shapiro test we observed a non-normal distribution for X1stFlrSF (p-value < 2.2e-16). Afterwards, we computed the InterQuartileRange to build the thresholds for severe outliers. 20 outliers were observed,from which 3 of them were severe outliers.

```r
summary(house_prices$X1stFlrSF)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     334     882    1087    1163    1391    4692
```

```r
# Histogram plotting


# Missing values
sum(is.na(house_prices$X1stFlrSF))
```

```
## [1] 0
```

```r
# Checking for normal distribution
shapiro.test(house_prices$X1stFlrSF)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$X1stFlrSF
## W = 0.92695, p-value < 2.2e-16
```

variable 11: X2ndFlrSF

X2ndFlrSF is a numerical variable with no missing values. 0 correspond to houses which do not have second floor. Then we used a histogram and a Boxplot to visualize the distribution of the values of this variable. By using a Shapiro test we observed a non-normal distribution for X2ndFlrSF (p-value < 2.2e-16). Afterwards, we computed the InterQuartileRange to build the thresholds for severe outliers. 2 outliers were observed,from which none of them were severe outliers.

```r
summary(house_prices$X2ndFlrSF)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0     347     728    2065
```

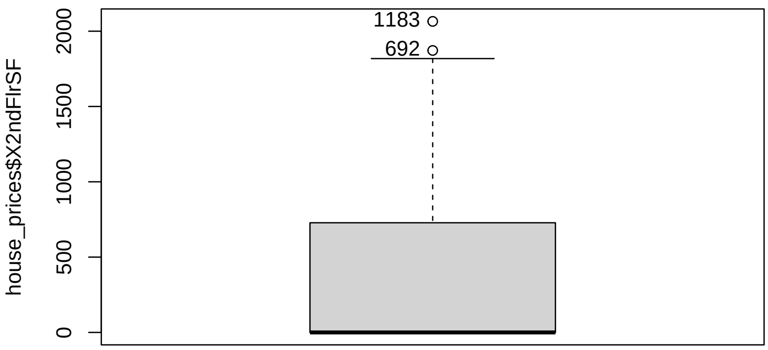
```r
# Missing values
sum(is.na(house_prices$X2ndFlrSF))
```

```
## [1] 0
```

```r
# Checking for normal distribution
shapiro.test(house_prices$X2ndFlrSF)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$X2ndFlrSF
## W = 0.7668, p-value < 2.2e-16
```

```r
# Univariant Outliers
length(Boxplot(house_prices$X2ndFlrSF, id = list(n = Inf)))
```




```
## [1] 2
```

```r
# Boxplot(house_prices$X2ndFlrSF, id = list(n = Inf))
sev_up <- (quantile(house_prices$X2ndFlrSF, 0.75) + (3 * ((quantile(house_prices$X2ndFlrSF, 0.75) - quantile(hou
sev_down <- (quantile(house_prices$X2ndFlrSF, 0.25) - (3 * ((quantile(house_prices$X2ndFlrSF, 0.75) - quantile(h
length(which(house_prices$X2ndFlrSF > sev_up))
```

```
## [1] 0
```

```r
length(which(house_prices$X2ndFlrSF < sev_down))
```

```
## [1] 0
```

variable 12: LowQualFinSF

LowQualFinSF is a numerical variable with no missing values. 0 correspond to houses with high quality of finished square feet. Then we used a histogram and a Boxplot to visualize the distribution of the values of this variable. By using a Shapiro test we observed a non-normal distribution for LowQualFinSF (p-value < 2.2e-16). 26 outliers were observed, all of them were the rows which had values (the rest were 0).

```r
summary(house_prices$LowQualFinSF)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   5.845   0.000 572.000
```

```r
# Missing values
sum(is.na(house_prices$LowQualFinSF))
```

```
## [1] 0
```

```r
# Checking for normal distribution
shapiro.test(house_prices$LowQualFinSF)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$LowQualFinSF
## W = 0.09799, p-value < 2.2e-16
```

variable 13: GrLivArea

GrLivArea is a numerical variable with no missing values. Then we used a histogram and a Boxplot to visualize the distribution of the values of this variable. By using a Shapiro test we observed a non-normal distribution for GrLivArea (p-value < 2.2e-16). 31 outliers were observed, from which only 4 were observed to be severe.

```
summary(house_prices$GrLivArea)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     334    1130    1464    1515    1777    5642
```

```
# Missing values
sum(is.na(house_prices$GrLivArea))
```

```
## [1] 0
```

```
# Checking for normal distribution
shapiro.test(house_prices$GrLivArea)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$GrLivArea
## W = 0.92798, p-value < 2.2e-16
```

variable 14: BsmtFullBath

BsmtFullBath is a numerical variable but contains only 4 possible values. Here we decided to categorize it with as.factor(). Then we used a barplot to visualize the distribution of the values of this variable. No missings were observed.

```
summary(house_prices$BsmtFullBath)
```

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

```
house_prices$BsmtFullBath <- as.factor(house_prices$BsmtFullBath)

# Missing values
sum(is.na(house_prices$BsmtFullBath))
```

```
## [1] 0
```

variable 15: BsmtHalfBath

BsmtHalfBath is a numerical variable but contains only 3 possible values. Here we decided to categorize it with as.factor(). Then we used a barplot to visualize the distribution of the values of this variable. No missings were observed.

```
summary(house_prices$BsmtHalfBath)
```

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

```
house_prices$BsmtHalfBath <- as.factor(house_prices$BsmtHalfBath)

# Missing values
sum(is.na(house_prices$BsmtHalfBath))
```

```
## [1] 0
```

variable 16: FullBath

FullBath is a numerical variable but contains only 4 possible values. Here we decided to categorize it with as.factor(). Then we used a barplot to visualize the distribution of the values of this variable. No missings were observed.

```
summary(house_prices$FullBath)
```

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

```
house_prices$FullBath <- as.factor(house_prices$FullBath)

# Missing values
sum(is.na(house_prices$FullBath))
```

```
## [1] 0
```

variable 17: HalfBath

HalfBath is a numerical variable but contains only 3 possible values. Here we decided to categorize it with as.factor(). Then we used a barplot to visualize the distribution of the values of this variable. No missings were observed.

```r
summary(house_prices$HalfBath)
```

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

```r
house_prices$HalfBath <- as.factor(house_prices$HalfBath)

# Missing values
sum(is.na(house_prices$HalfBath))
```

```
## [1] 0
```

variable 18: BedroomAbvGr

BedroomAbvGr is a numerical variable but contains only 9 possible values. Here we decided to categorize it with as.factor(). Then we used a barplot to visualize the distribution of the values of this variable. No missings were observed.

```r
summary(house_prices$BedroomAbvGr)
```

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

```r
house_prices$BedroomAbvGr <- as.factor(house_prices$BedroomAbvGr)

# Missing values
sum(is.na(house_prices$BedroomAbvGr))
```

```
## [1] 0
```

variable 19: KitchenAbvGr

KitchenAbvGr is a numerical variable but contains only 4 possible values. Here we decided to categorize it with as.factor(). Then we used a barplot to visualize the distribution of the values of this variable. No missings were observed.

```r
summary(house_prices$KitchenAbvGr)
```

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

```r
house_prices$KitchenAbvGr <- as.factor(house_prices$KitchenAbvGr)

# Missing values
sum(is.na(house_prices$KitchenAbvGr))
```

```
## [1] 0
```

variable 20: TotRmsAbvGrd

KitchenAbvGr is a numerical variable but contains only 12 possible values. Here we decided to categorize it with as.factor(). Then we used a barplot to visualize the distribution of the values of this variable. No missings were observed.

```r
summary(house_prices$TotRmsAbvGrd)
```

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

```r
house_prices$TotRmsAbvGrd <- as.factor(house_prices$TotRmsAbvGrd)

# Missing values
sum(is.na(house_prices$TotRmsAbvGrd))
```

```
## [1] 0
```

variable 21: Fireplaces

Fireplaces is a numerical variable but contains only 4 possible values. Here we decided to factorize it with as.factor(). Then we used a barplot to visualize the distribution of the values of this variable. No missings were observed.

```r
summary(house_prices$Fireplaces)
```

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

```r
house_prices$Fireplaces <- as.factor(house_prices$Fireplaces)

# Missing values
sum(is.na(house_prices$Fireplaces))
```

```
## [1] 0
```

variable 22: GarageYrBlt

GarageYrBlt is a numeric interval variable. It contains 81 NA's, that correspond to the houses with no garages. By using a Shapiro test we observed a non-normal distribution for YearBuilt (p-value < 2.2e-16). Afterwards, we computed the InterQuartileRange to build the thresholds for severe outliers. 0 outliers were seen in this variable.

```r
summary(house_prices$GarageYrBlt)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    1900    1961    1980    1979    2002    2010      81
```

```r
# Missing values
sum(is.na(house_prices$GarageYrBlt))
```

```
## [1] 81
```

```r
# Checking for normal distribution
shapiro.test(house_prices$GarageYrBlt)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$GarageYrBlt
## W = 0.92094, p-value < 2.2e-16
```

variable 23: GarageCars

This is a discrete quantitative variable, with only 5 values. It contains no missing values thus imputation is not needed. The variable contains 5 outliers (out of which 0 severe), all on the higher end of the spectrum.

```r
summary(house_prices$GarageCars)
```

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

```r
# Missing values
sum(is.na(house_prices$GarageCars))
```

```
## [1] 0
```

variable 24: GarageArea

This is a continuous ratio variable. The data is not normally distributed, which is confirmed by the near-null p-value of the shapiro normallity test. It contains no missing values thus imputation is not needed. The variable contains 21 outliers (out of which 3 severe), all on the higher end of the spectrum.

```r
summary(house_prices$GarageArea)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0   334.5   480.0   473.0   576.0  1418.0
```

```r
shapiro.test(house_prices$GarageArea)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$GarageArea
## W = 0.97533, p-value = 4.017e-15
```

```r
# Missing values
sum(is.na(house_prices$GarageArea))
```

```
## [1] 0
```

variable 25: WoodDeckSF

This is a continuous ratio variable. The data is not normally distributed, which is confirmed by the near-null p-value of the shapiro normallity test. It contains no missing values thus imputation is not needed. The variable contains 32 outliers (out of which 3 severe), all on the higher end of the spectrum.

```r
summary(house_prices$WoodDeckSF)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00   94.24  168.00  857.00
```

```r
shapiro.test(house_prices$WoodDeckSF)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$WoodDeckSF
## W = 0.76852, p-value < 2.2e-16
```

```r
# Missing values
sum(is.na(house_prices$WoodDeckSF))
```

```
## [1] 0
```

variable 26: OpenPorchSF

This is a continuous ratio variable. The data is not normally distributed, which is confirmed by the near-null p-value of the shapiro normallity test. It contains no missing values thus imputation is not needed. The variable contains 77 outliers (out of which 18 severe), all on the higher end of the spectrum.

```r
summary(house_prices$OpenPorchSF)
```

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

```r
shapiro.test(house_prices$OpenPorchSF)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$OpenPorchSF
## W = 0.72717, p-value < 2.2e-16
```

```r
# Missing values
sum(is.na(house_prices$OpenPorchSF))
```

```
## [1] 0
```

variable 27: EnclosedPorch

This is a continuous ratio variable. The data is not normally distributed, which is confirmed by the near-null p-value of the shapiro normallity test. It contains no missing values thus imputation is not needed. The variable contains 208 outliers (out of which 208 severe). This occurs because the majority of houses don't have an enclosed porch, so any house with an enclosed porch is considered an outlier.

```r
summary(house_prices$EnclosedPorch)
```

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

```r
shapiro.test(house_prices$EnclosedPorch)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$EnclosedPorch
## W = 0.41444, p-value < 2.2e-16
```

```r
# Missing values
sum(is.na(house_prices$EnclosedPorch))
```

```
## [1] 0
```

variable 28: X3SsnPorch

This is a continuous ratio variable. The data is not normally distributed, which is confirmed by the near-null p-value of the shapiro normallity test. It contains no missing values thus imputation is not needed. The variable contains 24 outliers (out of which 24 severe). This occurs because the majority of houses don't have a three season porch, so any house with a three season porch is considered an outlier.

```r
summary(house_prices$X3SsnPorch)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00    3.41    0.00  508.00
```

```r
shapiro.test(house_prices$X3SsnPorch)
```

```
##
##  Shapiro-Wilk normality test
##
```

```
## data:  house_prices$X3SsnPorch
## W = 0.094934, p-value < 2.2e-16
```

```r
# Missing values
sum(is.na(house_prices$X3SsnPorch))
```

```
## [1] 0
```

variable 29: ScreenPorch

This is a continuous ratio variable. The data is not normally distributed, which is confirmed by the near-null p-value of the shapiro normallity test. It contains no missing values thus imputation is not needed. The variable contains 116 outliers (out of which 116 severe). This occurs because the majority of houses don't have a screen porch, so any house with a screen porch is considered an outlier.

```r
summary(house_prices$ScreenPorch)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00   15.06    0.00  480.00
```

```r
shapiro.test(house_prices$ScreenPorch)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$ScreenPorch
## W = 0.29821, p-value < 2.2e-16
```

```r
# Missing values
sum(is.na(house_prices$ScreenPorch))
```

```
## [1] 0
```

variable 30: PoolArea

This is a continuous ratio variable. The data is not normally distributed, which is confirmed by the near-null p-value of the shapiro normallity test. It contains no missing values thus imputation is not needed. The variable contains 7 outliers (out of which 7 severe). This occurs because the majority of houses don't have a pool, so any house with a pool is considered an outlier.

```r
summary(house_prices$PoolArea)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   2.759   0.000 738.000
```

```r
shapiro.test(house_prices$PoolArea)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$PoolArea
## W = 0.041202, p-value < 2.2e-16
```

```r
# Missing values
sum(is.na(house_prices$PoolArea))
```

```
## [1] 0
```

variable 31: MiscVal

This is a continuous ratio variable. The data is not normally distributed, which is confirmed by the near-null p-value of the shapiro normallity test. It contains no missing values thus imputation is not needed. The variable contains 52 outliers (out of which 52 severe). This occurs because the majority of houses don't have miscellaneous features, so any house with a miscellaneous feature is considered an outlier.

```r
summary(house_prices$MiscVal)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     0.00     0.00     0.00    43.49     0.00 15500.00
```

```r
shapiro.test(house_prices$MiscVal)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$MiscVal
## W = 0.058233, p-value < 2.2e-16
```

```r
# Missing values
sum(is.na(house_prices$MiscVal))
```

```
## [1] 0
```

```r
# Imputing missing values
# res.pca<-imputePCA(house_prices[,c(2:)])
```

variable 32: MoSold

This is an ordinal categorical variable. The data is not normally distributed, which is confirmed by the near-null p-value of the shapiro normallity test. It contains no missing values thus imputation is not needed. The variable contains no outliers.

```r
summary(house_prices$MoSold)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   5.000   6.000   6.322   8.000  12.000
```

```r
shapiro.test(house_prices$MoSold)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$MoSold
## W = 0.96878, p-value < 2.2e-16
```

```r
# Missing values
sum(is.na(house_prices$MoSold))
```

```
## [1] 0
```

variable 33: YrSold

This is a discrete numerical variable. The data is not normally distributed, which is confirmed by the near-null p-value of the shapiro normallity test. It contains no missing values thus imputation is not needed. The variable contains no outliers.

```r
summary(house_prices$YrSold)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2006    2007    2008    2008    2009    2010
```

```r
shapiro.test(house_prices$YrSold)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$YrSold
## W = 0.8971, p-value < 2.2e-16
```

```r
# Missing values
sum(is.na(house_prices$YrSold))
```

```
## [1] 0
```

variable 34: SalePrice

This is a continuous ratio variable. The data is not normally distributed, which is confirmed by the near-null p-value of the shapiro normallity test, but this fact is further answered. It contains no missing values thus imputation is not needed. The variable contains 61 outliers (out of which 12 severe), all on the higher end of the spectrum.

```r
summary(house_prices$SalePrice)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   34900  129975  163000  180921  214000  755000
```

```r
shapiro.test(house_prices$SalePrice)
```

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

```r
# Missing values
sum(is.na(house_prices$SalePrice))
```

```
## [1] 0
```

variable 35: OverallQual

This is an ordinal categorical variable with 10 levels in which "Very Poor", "Poor", "Fair" and "Very Excellent" represent less than 3% of the instances combined. It contains no missing values thus imputation is not needed. A bar plot is used to plot the variable.

```
summary(house_prices$OverallQual)
```

```
##      Very Poor           Poor           Fair  Below Average        Average
##              2              3             20            116            397
##  Above Average           Good      Very Good      Excellent Very Excellent
##            374            319            168             43             18
```

```
prop.table(table(house_prices$OverallQual))
```

```
##
##      Very Poor           Poor           Fair  Below Average        Average
##    0.001369863    0.002054795    0.013698630    0.079452055    0.271917808
##  Above Average           Good      Very Good      Excellent Very Excellent
##    0.256164384    0.218493151    0.115068493    0.029452055    0.012328767
```

```
# Missing values
sum(is.na(house_prices$OverallQual))
```

```
## [1] 0
```

variable 36: Neighborhood

This is a nominal categorical variable (with 25 levels), in which "College Creek", "Edwards", "North Ames" and "Old Town" represent approximately 40% of the instances combined. It contains no missing values thus imputation is not needed. A bar plot is used to plot the variable.

```
prop.table(table(house_prices$Neighborhood))
```

```
##
##                   Bloomington Heights                              Bluestem
##                           0.011643836                           0.001369863
##                             Briardale                             Brookside
##                           0.010958904                           0.039726027
##                           Clear Creek                         College Creek
##                           0.019178082                           0.102739726
##                              Crawford                               Edwards
##                           0.034931507                           0.068493151
##                               Gilbert              Iowa DOT and Rail Road
##                           0.054109589                           0.025342466
##                        Meadow Village                              Mitchell
##                           0.011643836                           0.033561644
##                            North Ames                            Northridge
##                           0.154109589                           0.028082192
##                       Northpark Villa                   Northridge Heights
##                           0.006164384                           0.052739726
##                        Northwest Ames                              Old Town
##                           0.050000000                           0.077397260
## South & West of Iowa State University                                Sawyer
##                           0.017123288                           0.050684932
##                           Sawyer West                              Somerset
##                           0.040410959                           0.058904110
##                           Stone Brook                            Timberland
##                           0.017123288                           0.026027397
##                               Veenker
##                           0.007534247
```

```
# Missing values
sum(is.na(house_prices$Neighborhood))
```

```
## [1] 0
```

variable 37: ExterQual

This is a nominal categorical variable (with 5 levels), in which 62% of the instances are "Average/Typical" and 33% are "Good". The "Poor" level has 0 instances. It contains no missing values thus imputation is not needed. A bar plot is used to plot the variable.

```
summary(house_prices$ExterQual)
```

```
##       Excellent            Good Average/Typical            Fair            Poor
##              52             488             906              14               0
```

```r
prop.table(table(house_prices$ExterQual))
```

```
##
##       Excellent            Good Average/Typical            Fair            Poor
##     0.035616438     0.334246575     0.620547945     0.009589041     0.000000000
```

```r
# Missing values
sum(is.na(house_prices$ExterQual))
```

```
## [1] 0
```

variable 38: BsmtQual

This is a nominal categorical variable (with 6 levels), in which 42% of the instances are "Good (90-99 inches)" and 44% are "Typical (80-89 inches)". The "Poor (<70 inches)" level has 0 instances, and 2.5% of houses don't have a basement. A bar plot is used to plot the variable.

```r
table(house_prices$BsmtQual)
```

```
##
## Excellent (100+ inches)     Good (90-99 inches)  Typical (80-89 inches)
##                     121                     618                     649
##     Fair (70-79 inches)       Poor (<70 inches)             No Basement
##                      35                       0                      37
```

```r
prop.table(table(house_prices$BsmtQual))
```

```
##
## Excellent (100+ inches)     Good (90-99 inches)  Typical (80-89 inches)
##              0.08287671              0.42328767              0.44452055
##     Fair (70-79 inches)       Poor (<70 inches)             No Basement
##              0.02397260              0.00000000              0.02534247
```

```r
# Missing values ----> añadir si no se ha hecho antes
sum(is.na(house_prices$BsmtQual))
```

```
## [1] 0
```

variable 39: KitchenQual

This is a nominal categorical variable (with 5 levels), in which 40% of the instances are "Good" and 50% are "Typical/Average". The "Poor" level has 0 instances. It contains no missing values thus imputation is not needed. A bar plot is used to plot the variable.

```r
table(house_prices$KitchenQual)
```

```
##
##       Excellent            Good Typical/Average            Fair            Poor
##             100             586             735              39               0
```

```r
prop.table(table(house_prices$KitchenQual))
```

```
##
##       Excellent            Good Typical/Average            Fair            Poor
##      0.06849315      0.40136986      0.50342466      0.02671233      0.00000000
```

```r
# Missing values
sum(is.na(house_prices$KitchenQual))
```

```
## [1] 0
```

variable 40: GarageFinish

This is a nominal categorical variable (with 4 levels). It is visualized by a bar plot, in which houses with no garage represent only 5.5% of the instances.

```r
table(house_prices$GarageFinish)
```

```
##
##       Finished Rough Finished     Unfinished      No Garage
##            352            422            605             81
```

```r
prop.table(table(house_prices$GarageFinish))
```

```
##
##       Finished Rough Finished     Unfinished      No Garage
##     0.24109589     0.28904110     0.41438356     0.05547945
```

```r
# Missing values ----> añadir si no se ha hecho antes
sum(is.na(house_prices$GarageFinish))
```

```
## [1] 0
```

variable 41: FireplaceQu

This is a nominal categorical variable (with 6 levels), in which 49% of the instances are "Good" and 41% are "Average". The "Poor" level has 20 instances (2.6%). 47% of the houses have no fireplace.

```r
table(house_prices$FireplaceQu)
```

```
##
##    Excellent         Good      Average         Fair         Poor No Fireplace
##           24          380          313           33           20          690
```

```r
prop.table(table(house_prices$FireplaceQu))
```

```
##
##    Excellent         Good      Average         Fair         Poor No Fireplace
##   0.01643836   0.26027397   0.21438356   0.02260274   0.01369863   0.47260274
```

```r
# Missing values
sum(is.na(house_prices$FireplaceQu))
```

```
## [1] 0
```

variable 42: Foundation

This is a nominal categorical variable (with 6 levels), in which 43% of the instances are "Cinder Block" and 44% are "Poured Contrete". "Wood", "Stone" and "Slab" levels combined represent only 2.2% of the instances. It contains no missing values thus imputation is not needed. A bar plot is used to plot the variable.

```r
table(house_prices$Foundation)
```

```
##
##    Brick & Tile    Cinder Block Poured Contrete            Slab           Stone
##             146             634             647              24               6
##            Wood
##               3
```

```r
prop.table(table(house_prices$Foundation))
```

```
##
##    Brick & Tile    Cinder Block Poured Contrete            Slab           Stone
##     0.100000000     0.434246575     0.443150685     0.016438356     0.004109589
##            Wood
##     0.002054795
```

```r
# Missing values
sum(is.na(house_prices$Foundation))
```

```
## [1] 0
```

variable 43: GarageType

This is a nominal categorical variable (with 7 levels), in which 60% of the instances are "Attached to home" and 27% are "Detached from home". "More than one type of garage", "Basement Garage" and "Car Port" levels combined represent only 2.3% of the instances. 5.5% of the houses have no garage

```r
table(house_prices$GarageType)
```

```
##
##                                        More than one type of garage
##                                                                   6
##                                                    Attached to home
##                                                                 870
##                                                     Basement Garage
##                                                                  19
## Built-In (Garage part of house - typically has room above garage)
##                                                                  88
##                                                            Car Port
##                                                                   9
##                                                  Detached from home
##                                                                 387
##                                                           No Garage
```

```
##                                                           81
```

```r
prop.table(table(house_prices$GarageType))
```

```
## 
##                                      More than one type of garage 
##                                                       0.004109589 
##                                                  Attached to home 
##                                                       0.595890411 
##                                                   Basement Garage 
##                                                       0.013013699 
## Built-In (Garage part of house - typically has room above garage) 
##                                                       0.060273973 
##                                                          Car Port 
##                                                       0.006164384 
##                                                Detached from home 
##                                                       0.265068493 
##                                                         No Garage 
##                                                       0.055479452
```

```r
# Missing values
sum(is.na(house_prices$GarageType))
```

```
## [1] 0
```

variable 44: MSSubClass

This is a nominal categorical variable (with 16 levels), in which 37% of the instances are "1-STORY 1946 & NEWER ALL STYLES" and 20% are "2-STORY 1946 & NEWER". "1-STORY W/FINISHED ATTIC ALL AGES", "PUD - MULTILEVEL - INCL SPLIT LEV/FOYER" and "1-1/2 STORY - UNFINISHED ALL AGES" levels combined represent less than 2% of the instances. It contains no missing values thus imputation is not needed. A bar plot is used to plot the variable.

```r
prop.table(table(house_prices$MSSubClass))
```

```
## 
##                      1-STORY 1946 & NEWER ALL STYLES 
##                                          0.367123288 
##                                 1-STORY 1945 & OLDER 
##                                          0.047260274 
##                    1-STORY W/FINISHED ATTIC ALL AGES 
##                                          0.002739726 
##                    1-1/2 STORY - UNFINISHED ALL AGES 
##                                          0.008219178 
##                        1-1/2 STORY FINISHED ALL AGES 
##                                          0.098630137 
##                                 2-STORY 1946 & NEWER 
##                                          0.204794521 
##                                 2-STORY 1945 & OLDER 
##                                          0.041095890 
##                                 2-1/2 STORY ALL AGES 
##                                          0.010958904 
##                                 SPLIT OR MULTI-LEVEL 
##                                          0.039726027 
##                                          SPLIT FOYER 
##                                          0.013698630 
##                         DUPLEX - ALL STYLES AND AGES 
##                                          0.035616438 
## 1-STORY PUD (Planned Unit Development) - 1946 & NEWER 
##                                          0.059589041 
##                             1-1/2 STORY PUD - ALL AGES 
##                                          0.000000000 
##                             2-STORY PUD - 1946 & NEWER 
##                                          0.043150685 
##              PUD - MULTILEVEL - INCL SPLIT LEV/FOYER 
##                                          0.006849315 
##            2 FAMILY CONVERSION - ALL STYLES AND AGES 
##                                          0.020547945
```

```r
# Missing values
sum(is.na(house_prices$MSSubClass))
```

```
## [1] 0
```

2 Data quality report

2.1 Missing Imputation

We have three variables with missing values (GarageYrBlt, LotFrontage, MasVnrArea). - GarageYrBlt has 81 NAs, which correspond to the 81 houses with no garage. Thus, it is impossible to impute a value for these missings. We thought of assigning a sentinel value as 0 to these missing values, but this could affect the imputation of the other variables' missing values. As the correlation between the variables GarageYrBlt and YearBuilt is significantly high (0.83, indicating multicollinearity), and the correlation test returns a near-null p-value, we decided to delete the variable. - For the other two variables, the missing values are random, so we decided to use the imputePCA algorithm for imputation. We observed that the imputations haven't changed the dataset significantly.

```
# GarageYrBlt
ll <- which(is.na(house_prices$GarageYrBlt))

testdf <- house_prices[-ll, ]
cor.test(testdf$YearBuilt, testdf$GarageYrBlt)
```

```
##
##  Pearson's product-moment correlation
##
## data:  testdf$YearBuilt and testdf$GarageYrBlt
## t = 54.309, df = 1377, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.8081008 0.8417668
## sample estimates:
##       cor
## 0.8256675
```

```
house_prices <- subset(house_prices, select = -GarageYrBlt)
# LotFrontage, MasVnrArea
res.pca <- imputePCA(house_prices[, c(2:14, 24:34)]) # Imputation for numeric variables only
house_prices$LotFrontage <- res.pca$completeObs[, 1]
house_prices$MasVnrArea <- res.pca$completeObs[, 5]
```

2.2 Observed relations

Strong Positive Correlations among Numerical Features (> 0.45): - GarageCars and GarageArea (0.85) - X1stFlrSF and TotalBsmtSF (0.83) - LotFrontage and LotArea (0.60) - YearBuilt and GarageArea (0.54) - GrLivArea and X1stFlrSF (0.47) - GrLivArea and X2ndFlrSF (0.64) - GarageArea and X1stFlrSF (0.48) Negative Correlations among Features (< -0.40): - BsmtUnfSF and BsmtFinSF1 (-0.58) - EnclosedPorch and YearBuilt (-0.41)

```
# cor(house_prices[, c(2:14, 23:34)], method = "spearman")
```

2.3 Univariate Outliers

Now the individuals are investigated. First the number of univariate outliers per individual are counted and added in a new variable called 'univ_outl_count'. Looking at the 2 individuals with the most univariate outliers (>= 8) it can be concluded that they are all houses with a big living area and large LotArea. A correlation matrix confirms this as it shows a positive correlation to GrLiveArea, X1stFlrSF, LotArea and BsmtFinSF2.

```
house_prices$univ_outl_count <- 0
# List of numeric variables for which outliers are to be counted
numeric_variables <- c(
  "LotFrontage", "LotArea", "YearBuilt", "MasVnrArea", "BsmtFinSF1",
  "BsmtFinSF2", "BsmtUnfSF", "TotalBsmtSF", "X1stFlrSF", "X2ndFlrSF",
  "LowQualFinSF", "GrLivArea", "GarageCars", "GarageArea", "WoodDeckSF",
  "OpenPorchSF", "EnclosedPorch", "X3SsnPorch", "ScreenPorch", "PoolArea"
)
# Iterate through variables and update univ_outl_count
for (variable in numeric_variables) {
  variable_values <- house_prices[[variable]]
  variable_stats <- boxplot.stats(variable_values)
  outlier_indices <- which(variable_values %in% variable_stats$out)

  house_prices$univ_outl_count[outlier_indices] <- house_prices$univ_outl_count[outlier_indices] + 1
}
max(house_prices$univ_outl_count)
```

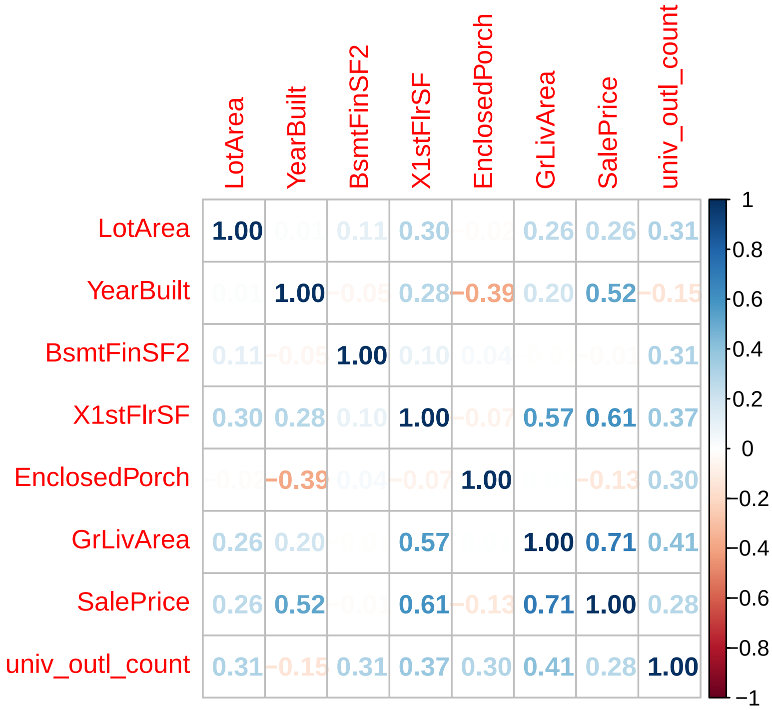
```
## [1] 10
```

```r
# house_prices[which(house_prices$univ_outl_count >= 8), ]

df_of_interest = house_prices[,c(3,4,8,11,27,14,34,45)]
cor_outl = cor(df_of_interest)
require(corrplot)
```

```
## Loading required package: corrplot
```

```
## corrplot 0.92 loaded
```

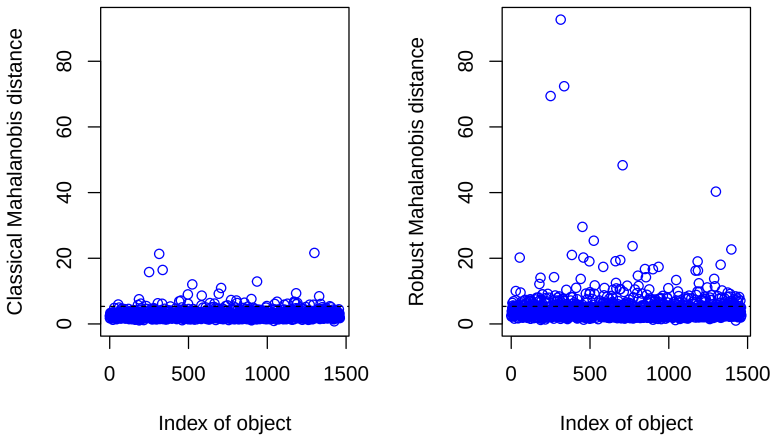
```r
par(mfrow = c(1, 1))
corrplot(cor_outl, method = "number")
```



2.4 Multivariate Outliers

Moutlier is applied on some numerical variables to find multivariate outliers. We chose the variables that don't return a singular matrix. A very mild threshold of 0.15% is chosen as significance level because it already returns a significant amount of outliers, more exactly around 3% of instances. It is chosen to delete these outliers from the data set for the rest of the project.

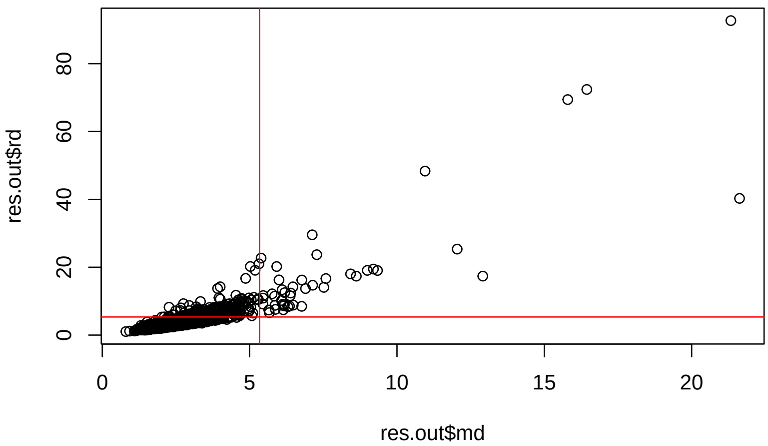
```r
res.out <- Moutlier(house_prices[, c(2, 3, 4, 7, 10, 14, 24, 26, 32, 34)], quantile = 0.9985, col = "blue")
```




```r
# which((res.out$md > res.out$cutoff) & (res.out$rd > res.out$cutoff))
length(which((res.out$md > res.out$cutoff) & (res.out$rd > res.out$cutoff))) / 1460
```

```
## [1] 0.03082192
```

```r
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")
```




```r
# summary(house_prices[which((res.out$md > res.out$cutoff) & (res.out$rd > res.out$cutoff)), ])
# summary(house_prices)
house_prices <- house_prices[-which((res.out$md > res.out$cutoff) & (res.out$rd > res.out$cutoff)), ]
```

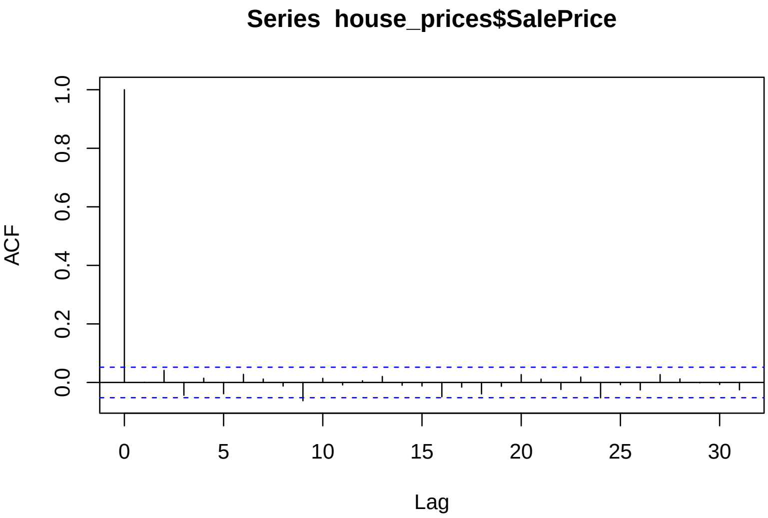
3 Profiling

3.1 Determine if the response variable (price) has an acceptably normal distribution. Address test to discard serial correlation.

The acf function in R plots the autocorrelation function of a time series, which measures the linear dependence of the series with itself at different lags. From this acf plot, we can conclude that the `SalePrice` variable does not exhibit any strong or consistent autocorrelation at different lags, and thus it is likely to be a random or stationary time series.

From the Shapiro test, we reject the null hypothesis and conclude that the `SalePrice` variable is not normally distributed.

```r
acf(house_prices$SalePrice)
```



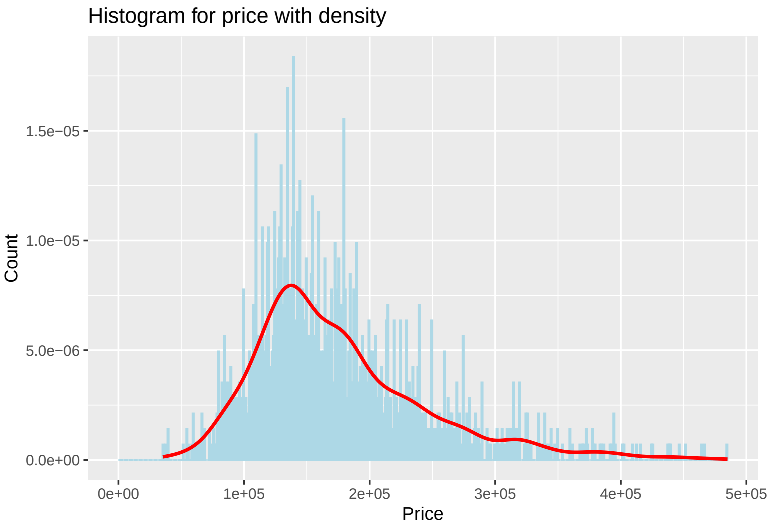

```r
shapiro.test(house_prices$SalePrice)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  house_prices$SalePrice
## W = 0.91999, p-value < 2.2e-16
```

```r
(
  ggplot(
    data = house_prices,
    aes(SalePrice, y = ..density..)
  ) +
    geom_histogram(
      breaks = seq(
        0,
        max(house_prices$SalePrice),
        by = 1000
      ),
      col = "lightblue",
      fill = "steelblue"
```

```
    ) +
    geom_density(
      lwd = 1,
      col = "red"
    ) +
    labs(
      title = "Histogram for price with density",
      x = "Price",
      y = "Count"
    )
)
```

```
## Warning: The dot-dot notation (`..density..`) was deprecated in ggplot2 3.4.0.
## i Please use `after_stat(density)` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



3.2 Categorize numeric variables.

Following the initial analysis on the numeric variables, these 3 columns have been categorized:

1. Linear feet of street connected to property.
2. Lot size in square feet.
3. Original construction date.

```
cols <- c("LotFrontage", "LotArea", "YearBuilt")
new_cols <- c("f.LotFrontage", "f.LotArea", "f.YearBuilt")
levels_list <- list(
  c(0, 59, 70, 80, 313), # LotFrontage
  c(0, 5000, 10000, 20000, 50000, 215245), # LotArea
  c(1872, 1915, 1945, 1960, 1980, 2000, 2010) # YearBuilt
)
labels_list <- list(
  c("Very Low", "Low", "Medium", "High"), # LotFrontage
  c("Small", "Medium", "Large", "Huge", "Very Huge"), # LotArea
  c(
    "Historic", "Pre-War", "Post-War", "Mid-Century",
    "Modern", "Contemporary"
  ) # YearBuilt
)

house_prices[new_cols] <- lapply(
  seq_along(cols),
  function(i) {
    factor(
      cut(
        house_prices[[cols[i]]],
        breaks = levels_list[[i]],
        labels = labels_list[[i]],
        right = TRUE,
        include.lowest = TRUE
```

```
      )
    )
  }
)
```

3.3 Interactions between categorical and numerical variables

Condes() is an R function from FactoMineR which is used to describe continuous by quantitative variables and/or by qualitative variables.

For quantitative variables, we observed 9 variables which showed a high correlation (Correlation > 50 & p-value around 0) with our target variable (SalePrice). These variables are : GrLivArea, GarageCars, GarageArea, TotalBsmtSF, X1stFlrSF, FullBath, TotRmsAbvGrd, YearBuilt & YearRemodAdd. Apparently, the house garages play an important role when determining the sale price. Also the size of the ground living area shows high significance in describing SalePrice. The age of the house (YearBuilt) and the remodelling (YearRemodAdd) is important to the target variable. On the other hand, we observed three variables which were negatively correlated with our target variable (EnclosedPorch, KitchenAbvGr & LowQualiFinSF).

For qualitative variables, three main features explained the most the variance in our target variable ($R2 > 0.5$ & p-value~0), which are OverallQual ($R2 = 0.70$ & p-value = 0), Neighborhood ($R2 = 0.59$ & p-value ~ 0) and ExterQual ($R2 = 0.51$ & p-value ~ 0). This is to be expected, as the quality of the materials used to build the house is significantly important to determine the SalePrice. The Neighborhood also explained most of the variance of the SalePrice, as expected. KitchenAbvGr, BedroomAbvGr, BsmtFullBath and HalfBath are poorly associated as they have R2-values under 10%.

```r
res.con <- condes(house_prices, num.var = 34)
# Assessing the description of the num variable by the quantitative variables
res.con$quanti
```

```
##                  correlation       p.value
## GrLivArea         0.71190414 3.892220e-219
## GarageCars        0.66128126 1.203708e-178
## GarageArea        0.64914213 4.386128e-170
## TotalBsmtSF       0.63469134 2.164739e-160
## X1stFlrSF         0.60389783 2.256494e-141
## YearBuilt         0.58818082 1.917365e-132
## YearRemodAdd      0.53915845 1.618933e-107
## MasVnrArea        0.46421085  1.556411e-76
## LotFrontage       0.39427800  7.731989e-54
## BsmtFinSF1        0.37688869  5.513029e-49
## OpenPorchSF       0.36899137  7.077661e-47
## LotArea           0.34415321  1.281555e-40
## WoodDeckSF        0.33246751  7.282340e-38
## X2ndFlrSF         0.28846918  1.605241e-28
## BsmtUnfSF         0.21955481  6.615046e-17
## univ_outl_count   0.12392464  2.932056e-06
## ScreenPorch       0.08404726  1.554199e-03
## MoSold            0.07910940  2.902727e-03
## X3SsnPorch        0.05884729  2.685696e-02
## LowQualFinSF     -0.07801155  3.320427e-03
## EnclosedPorch    -0.16124923  1.059161e-09
```

```r
# Assessing the description of the num variable by the quantitative variables
res.con$quali
```

```
##                       R2       p.value
## OverallQual   0.70447101  0.000000e+00
## Neighborhood  0.58508243 4.260331e-245
## ExterQual     0.50995636 6.188989e-218
## BsmtQual      0.49254865 6.997563e-206
## KitchenQual   0.46131304 5.850093e-189
## f.YearBuilt   0.40228000 1.259498e-154
## GarageFinish  0.35115286 5.208478e-132
## FullBath      0.32323825 4.025286e-119
## Foundation    0.30313329 7.462536e-108
## FireplaceQu   0.30227815 1.763095e-107
## GarageType    0.28756021 4.479010e-100
## TotRmsAbvGrd  0.28058615  2.564771e-93
## MSSubClass    0.28252170  1.019069e-90
## Fireplaces    0.23674025  2.457021e-82
## f.LotFrontage 0.20829301  3.755570e-71
## f.LotArea     0.14767071  1.258524e-47
```

```
## HalfBath      0.09361258  7.304544e-31
## BsmtFullBath  0.05729272  6.079005e-18
## BedroomAbvGr  0.04692003  1.176924e-12
## KitchenAbvGr  0.02205916  6.718825e-07
```

4 Price Modelling

4.1 Model building

4.2 Multicollinearity on the model

First, we built a model using only the numerical variables of our dataset. To simplify our model, collinearity is investigated to see if there are variables that are redundant in our model. We can see that there are some aliased coefficients, but it seems to be related to the interaction terms between certain variables. For fixing that, we decided to exclude TotalBsmtSF and GrLivArea. The TotalBsmtSF can be obtained adding (BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF), and the GrLivArea (X1stFlrSF + X2ndFlrSF + LowQualFinSF). We also excluded Id and univ_outl_count because they were either non-informative or were created artificially to perform a given section of the project (univ_outl_count).

Then we calculated the variance inflation factor. This indicates whether or not a variable correlates too much with other predictors such that it becomes redundant in the model. In general, a VIF-value larger than 1/(1-R_sq) is considered as showing too much collinear behavior. In our case, GarageCars has a value very close to the threshold as it is really correlated with the GarageArea, so we decided to exclude the GarageCars variable.

To further confirm this hypothesis, models are build by alternately removing the highly correlated variables from the logarithmic model. Then, ANOVA is applied to test whether or not the models are significantly predicting something else and AIC to see what model is considered the best. We remember that Strong Positive Correlations among Features (> 0.45): - GarageCars and GarageArea (0.85) - X1stFlrSF and TotalBsmtSF (0.83) - LotFrontage and LotArea (0.60) - YearBuilt and GarageArea (0.54) - GrLivArea and X1stFlrSF (0.47) - GrLivArea and X2ndFlrSF (0.64) - GarageArea and X1stFlrSF (0.48) Negative Correlations among Features (< -0.40): - BsmtUnfSF and BsmtFinSF1 (-0.58) - EnclosedPorch and YearBuilt (-0.41)

These tests show that the model with all numeric variables performs the best and that no severe collinearity is present in our model.

```r
numeric_variables <- sapply(house_prices, is.numeric)
m1 <- lm(SalePrice ~ ., data = house_prices[, numeric_variables])
# summary(m1)
# alias(m1)

# Creating another model without these variables
excluded <- c("Id", "TotalBsmtSF", "GrLivArea", "univ_outl_count")
selected <- numeric_variables & !names(numeric_variables) %in% excluded
m2 <- lm(SalePrice ~ ., data = house_prices[, selected])
t <- summary(m2)
# t
vif(m2)
```

```
##   LotFrontage       LotArea     YearBuilt  YearRemodAdd    MasVnrArea
##      1.535952      1.490310      2.477828      1.715328      1.314736
##    BsmtFinSF1    BsmtFinSF2     BsmtUnfSF     X1stFlrSF     X2ndFlrSF
##      4.156255      1.488468      4.105503      3.354766      1.532865
##  LowQualFinSF    GarageCars    GarageArea    WoodDeckSF   OpenPorchSF
##      1.037367      5.429126      5.101565      1.196251      1.207866
## EnclosedPorch    X3SsnPorch   ScreenPorch      PoolArea       MiscVal
##      1.242851      1.022488      1.071610      1.030334      1.010455
##        MoSold        YrSold
##      1.041770      1.043342
```

```r
1 / (1 - t$r.squared)
```

```
## [1] 5.580571
```

```r
excluded <- c("Id", "TotalBsmtSF", "GrLivArea", "univ_outl_count", "GarageCars")
selected <- numeric_variables & !names(numeric_variables) %in% excluded
m3 <- lm(SalePrice ~ ., data = house_prices[, selected])

excluded <- c("Id", "TotalBsmtSF", "GrLivArea", "univ_outl_count", "GarageCars", "LotArea")
selected <- numeric_variables & !names(numeric_variables) %in% excluded
m4 <- lm(SalePrice ~ ., data = house_prices[, selected])

excluded <- c("Id", "TotalBsmtSF", "GrLivArea", "univ_outl_count", "GarageCars", "YearBuilt")
selected <- numeric_variables & !names(numeric_variables) %in% excluded
m5 <- lm(SalePrice ~ ., data = house_prices[, selected])
```

```r
excluded <- c("Id", "TotalBsmtSF", "GrLivArea", "univ_outl_count", "GarageCars", "BsmtUnfSF")
selected <- numeric_variables & !names(numeric_variables) %in% excluded
m6 <- lm(SalePrice ~ ., data = house_prices[, selected])

anova(m3, m4)
```

```
## Analysis of Variance Table
##
## Model 1: SalePrice ~ LotFrontage + LotArea + YearBuilt + YearRemodAdd +
##     MasVnrArea + BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF + X1stFlrSF +
##     X2ndFlrSF + LowQualFinSF + GarageArea + WoodDeckSF + OpenPorchSF +
##     EnclosedPorch + X3SsnPorch + ScreenPorch + PoolArea + MiscVal +
##     MoSold + YrSold
## Model 2: SalePrice ~ LotFrontage + YearBuilt + YearRemodAdd + MasVnrArea +
##     BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF + X1stFlrSF + X2ndFlrSF +
##     LowQualFinSF + GarageArea + WoodDeckSF + OpenPorchSF + EnclosedPorch +
##     X3SsnPorch + ScreenPorch + PoolArea + MiscVal + MoSold +
##     YrSold
##   Res.Df        RSS Df   Sum of Sq      F    Pr(>F)
## 1   1393 1.2462e+12
## 2   1394 1.2614e+12 -1 -1.5251e+10 17.048 3.861e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(m3, m5)
```

```
## Analysis of Variance Table
##
## Model 1: SalePrice ~ LotFrontage + LotArea + YearBuilt + YearRemodAdd +
##     MasVnrArea + BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF + X1stFlrSF +
##     X2ndFlrSF + LowQualFinSF + GarageArea + WoodDeckSF + OpenPorchSF +
##     EnclosedPorch + X3SsnPorch + ScreenPorch + PoolArea + MiscVal +
##     MoSold + YrSold
## Model 2: SalePrice ~ LotFrontage + LotArea + YearRemodAdd + MasVnrArea +
##     BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF + X1stFlrSF + X2ndFlrSF +
##     LowQualFinSF + GarageArea + WoodDeckSF + OpenPorchSF + EnclosedPorch +
##     X3SsnPorch + ScreenPorch + PoolArea + MiscVal + MoSold +
##     YrSold
##   Res.Df        RSS Df   Sum of Sq     F    Pr(>F)
## 1   1393 1.2462e+12
## 2   1394 1.3343e+12 -1 -8.8066e+10 98.44 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
anova(m3, m6)
```

```
## Analysis of Variance Table
##
## Model 1: SalePrice ~ LotFrontage + LotArea + YearBuilt + YearRemodAdd +
##     MasVnrArea + BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF + X1stFlrSF +
##     X2ndFlrSF + LowQualFinSF + GarageArea + WoodDeckSF + OpenPorchSF +
##     EnclosedPorch + X3SsnPorch + ScreenPorch + PoolArea + MiscVal +
##     MoSold + YrSold
## Model 2: SalePrice ~ LotFrontage + LotArea + YearBuilt + YearRemodAdd +
##     MasVnrArea + BsmtFinSF1 + BsmtFinSF2 + X1stFlrSF + X2ndFlrSF +
##     LowQualFinSF + GarageArea + WoodDeckSF + OpenPorchSF + EnclosedPorch +
##     X3SsnPorch + ScreenPorch + PoolArea + MiscVal + MoSold +
##     YrSold
##   Res.Df        RSS Df   Sum of Sq      F    Pr(>F)
## 1   1393 1.2462e+12
## 2   1394 1.3257e+12 -1 -7.9461e+10 88.821 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
AIC(m3, m4, m5, m6)
```

```
##    df      AIC
## m3 23 33205.27
## m4 22 33220.48
```

```
## m5 22 33299.89
## m6 22 33290.73
```

The model's intercept was not statistically significant (p = 0.430036), suggesting that the predicted sale price is not significantly different from zero when all other predictors are zero. Among the predictor variables, several were statistically significant with positive or negative coefficients. For instance, variables such as "YearBuilt," "YearRemodAdd," "BsmtFinSF1," "X1stFlrSF," "X2ndFlrSF" and "GarageArea" had positive coefficients, indicating a positive relationship with sale price. On the other hand, the variable "YrSold" is -4.962e+02. Specifically, for each additional year the house was sold later, the SalePrice is expected to decrease by approximately 496.2 units. On average, more recent sales are associated with lower SalePrices. The overall model explained a substantial portion of the variability in sale prices (Adjusted R-squared = 0.8172), and the F-statistic was highly significant ($p < 2.2e\text{-}16$), indicating that at least one of the predictors was significantly related to the sale price. However, we observed that some of the predictors were not statistically significant, so we performed a Stepwise (step()) to remove them. By using step we were able to select a formula-based model by AIC. Here we were able to discard MiscVal, YrSold, X3SsnPorch, LowQualFinSF, PoolArea.

Afterwards we created a new model with the output model produced by step. In this model, we did observe a statistical significance for the intercept, suggesting a statistical significance from zero when the other predictors are zero. Almost all predictors showed a high significance in this model, so we kept all of them. At this point we attempted to incorporate categorical variables to our model.

As a last step to create our model, we introduced all our categorical variables to the model and we run again step() to remove non-significant predictors. Here we discarded LotFrontage, EnclosedPorch, MoSold, GarageFinish, FireplaceQu, Foundation, GarageType, f.LotFrontage, f.LotArea, BsmtFullBath, BsmtHalfBath, FullBath and TotRmsAbvGrd.

```r
excluded <- c("Id", "TotalBsmtSF", "GrLivArea", "univ_outl_count", "GarageCars")
selected <- numeric_variables & !names(numeric_variables) %in% excluded
m3 <- lm(SalePrice ~ ., data = house_prices[, selected])
# summary(m3)
# Now excluding no significant predictors
step(m3, trace = 0)
```

```
##
## Call:
## lm(formula = SalePrice ~ LotFrontage + LotArea + YearBuilt +
##     YearRemodAdd + MasVnrArea + BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF +
##     X1stFlrSF + X2ndFlrSF + GarageArea + WoodDeckSF + OpenPorchSF +
##     EnclosedPorch + ScreenPorch + MoSold, data = house_prices[,
##     selected])
##
## Coefficients:
##   (Intercept)    LotFrontage        LotArea      YearBuilt   YearRemodAdd
##    -1.950e+06      9.390e+01      8.780e-01      4.039e+02      5.813e+02
##    MasVnrArea     BsmtFinSF1     BsmtFinSF2      BsmtUnfSF      X1stFlrSF
##     3.267e+01      5.393e+01      3.023e+01      3.461e+01      5.617e+01
##     X2ndFlrSF     GarageArea     WoodDeckSF    OpenPorchSF  EnclosedPorch
##     6.022e+01      4.415e+01      2.518e+01      5.172e+01      3.475e+01
##   ScreenPorch         MoSold
##     4.494e+01      5.071e+02
```

```r
# Model with the subselection of variables
m7 <- lm(formula = SalePrice ~ LotFrontage + LotArea + YearBuilt +
  YearRemodAdd + MasVnrArea + BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF +
  X1stFlrSF + X2ndFlrSF + GarageArea + WoodDeckSF + OpenPorchSF +
  EnclosedPorch + ScreenPorch + MoSold, data = house_prices)

# Adding categorical variables
m8 <- lm(formula = SalePrice ~ LotFrontage + LotArea + YearBuilt +
  YearRemodAdd + MasVnrArea + BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF +
  X1stFlrSF + X2ndFlrSF + GarageArea + WoodDeckSF + OpenPorchSF +
  EnclosedPorch + ScreenPorch + MoSold + ExterQual + BsmtQual +
  KitchenQual + GarageFinish + FireplaceQu + Foundation +
  GarageType + MSSubClass + Neighborhood + f.LotFrontage +
  f.LotArea + f.YearBuilt + OverallQual + BsmtFullBath +
  BsmtHalfBath + FullBath + HalfBath + BedroomAbvGr +
  KitchenAbvGr + TotRmsAbvGrd + Fireplaces, data = house_prices)
# step(m8, trace = 0)

# Final model
m9 <- lm(formula = SalePrice ~ LotArea + YearBuilt + YearRemodAdd +
  MasVnrArea + BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF + X1stFlrSF +
  X2ndFlrSF + GarageArea + WoodDeckSF + OpenPorchSF + ScreenPorch +
  ExterQual + BsmtQual + KitchenQual + MSSubClass + Neighborhood +
```

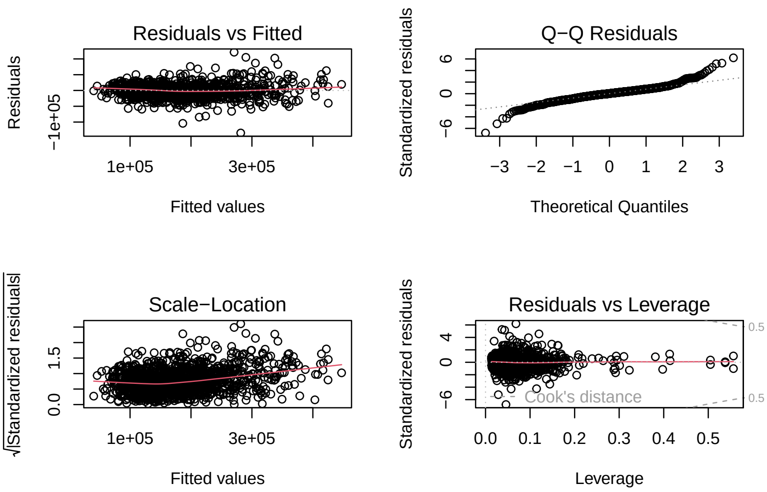
```
  f.YearBuilt + OverallQual + HalfBath + BedroomAbvGr + KitchenAbvGr +
  Fireplaces, data = house_prices)
# summary(m9)
```

5 Model validation

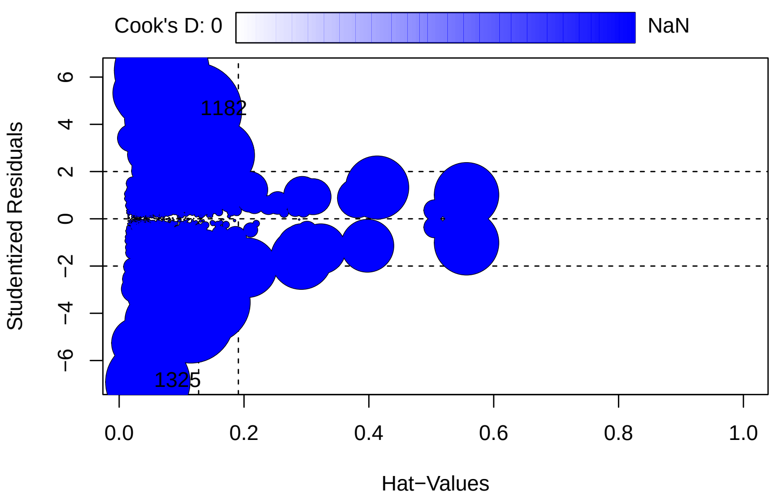
We used different approaches to validate if our model was correct or not. First of all we run diagnostic plots to our model using plot(). By looking into the Residuals vs Fitted plot, we observed homoscedasticity between residuals and fitted values (horizontal band), meaning that the variance of the residuals is constant across all levels of the independent variables. On the other hand, by looking into the Normal Q-Q plot, we observed that the residuals do not follow a complete normal distribution, as the ones in the 3 and -3 quantiles deviate from the straight line. Then we visualized the influence of each observation on the fitted values and residuals of our model. Here we observed that 1182, 1325 and 534 had a high influence on our residuals. Afterwards we plotted for each predictor of the model the response versus our data. We observed for every predictor homoscedasticity. We then used residualPlots() to plot residuals vs fitted for each predictor of our model. Again, we observed homoscedasticity. To assess the fit and assumptions of our regression model we used crPlots(). We observed linearity for all our predictors in our model. Finally, we used boxcox() to transform the response variable to a power of lambda, where lambda is a parameter that is determined such that the transformed variable follows a normal distribution.

```
library(MASS)
# Diagnostic plots for our model
par(mfrow = c(2, 2))
plot(m9, id.n = 0)
```

```
## Warning: not plotting observations with leverage one:
##   926
```

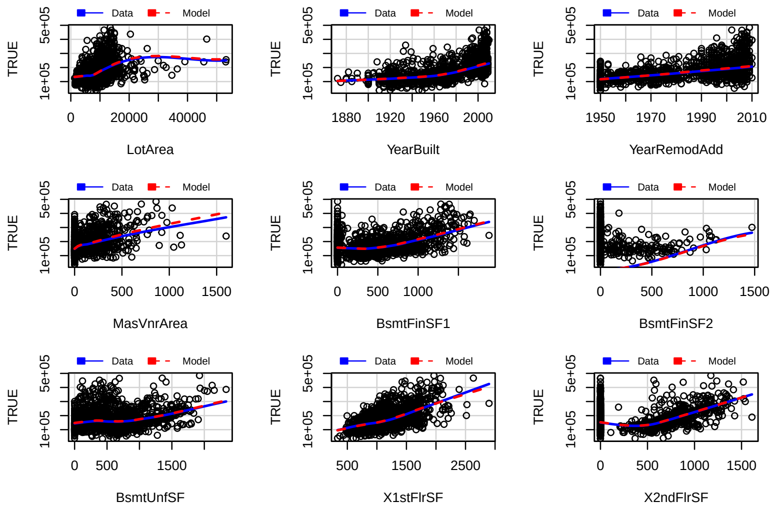



```
par(mfrow = c(1, 1))

# Influential data
influencePlot(m9, id = list(n = 0))
```

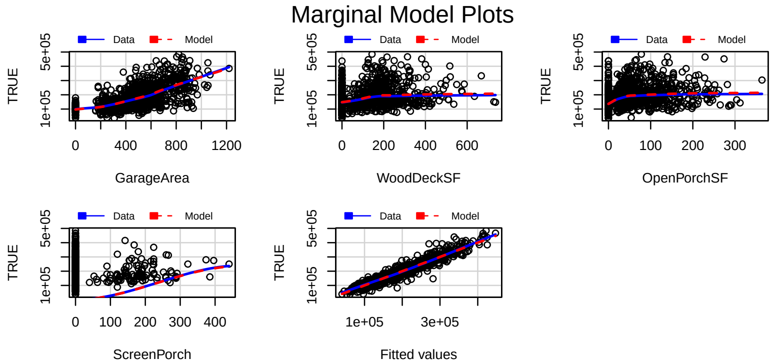



```
##           StudRes        Hat      CookD
## 955           NaN 1.00000000        NaN
## 1182   4.595600 0.12002096 0.03152693
## 1325 -6.912436 0.04591264 0.02467713
```

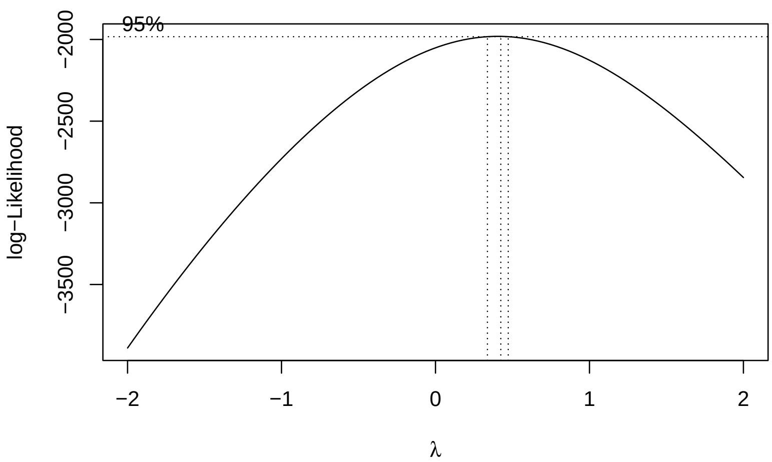
```r
# Marginal model plots
par(mfrow = c(2, 2))
marginalModelPlots(m9, id = list(n = 0))
```




```
## Warning in mmps(...): Interactions and/or factors skipped
```




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

# Residual plots+
# par(mfrow = c(2, 2))
# residualPlots(m9, id = list(n = 0))
# par(mfrow = c(1, 1))

# Component residual plots
# par(mfrow = c(2, 2))
# crPlots(m9, id = list(n = 0))
par(mfrow = c(1, 1))

# Boxcox
boxcox(SalePrice ~ LotArea + YearBuilt + YearRemodAdd +
  MasVnrArea + BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF + X1stFlrSF +
  X2ndFlrSF + GarageArea + WoodDeckSF + OpenPorchSF + ScreenPorch +
  ExterQual + BsmtQual + KitchenQual + MSSubClass + Neighborhood +
  f.YearBuilt + OverallQual + HalfBath + BedroomAbvGr + KitchenAbvGr +
  Fireplaces, data = house_prices)
```

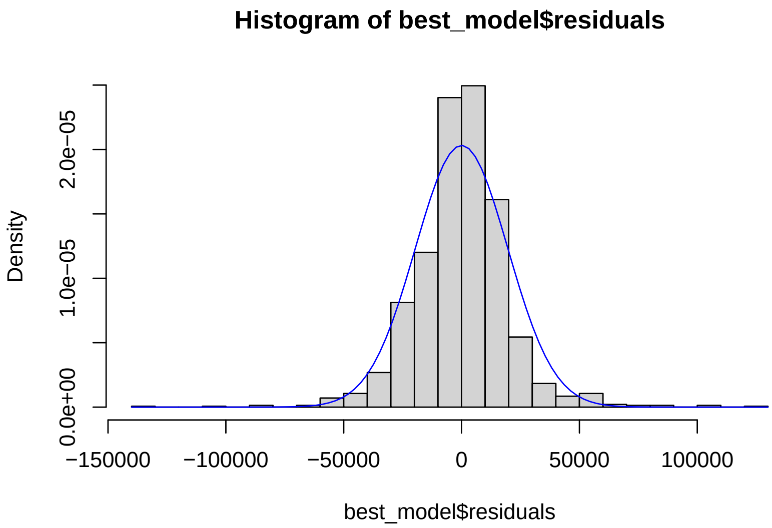


6 Residual outliers

The analysis shows that there are 16 residual outliers in the best model, which are the observations that have studentized residuals outside the 99% confidence interval. These outliers are shown in red in the boxplot, the residual plot, and the Cook's distance plot. The Cook's distance measures the influence of each observation on the fitted model, and the outliers have relatively high values, indicating that they have a large impact on the model. The summary of the outliers' data frame shows that these outliers have some extreme values or unusual combinations of the predictor variables.

Leveraging the `broom` library, the outliers with large positive or negative residuals are ploted, indicating the best model underestimates or overestimates the sale prive for them. For example, observation number 14 was underestimated by 67,000 dollars. It could correspond to large and luxurious house that have many features not captured by the model. On contrast, observation number 31 was overestimated by 58,000 dollars and could be due to a small and old house with many defects not captured by the model.

```
best_model <- m9
par(mfrow = c(1, 1))
hist(
  best_model$residuals,
  freq = FALSE,
  breaks = 20
)
curve(
  dnorm(
    x,
    mean(best_model$residuals),
    sd(best_model$residuals)
  ),
  col = "blue",
  add = T
)
```




```
residuals_lower_bound <- quantile(best_model$residuals, 0.005)
residuals_upper_bound <- quantile(best_model$residuals, 0.995)
residuals_outliers <- unname(which(
  best_model$residuals > residuals_upper_bound |
    best_model$residuals < residuals_lower_bound
))
length(residuals_outliers)
```

```
## [1] 16
```

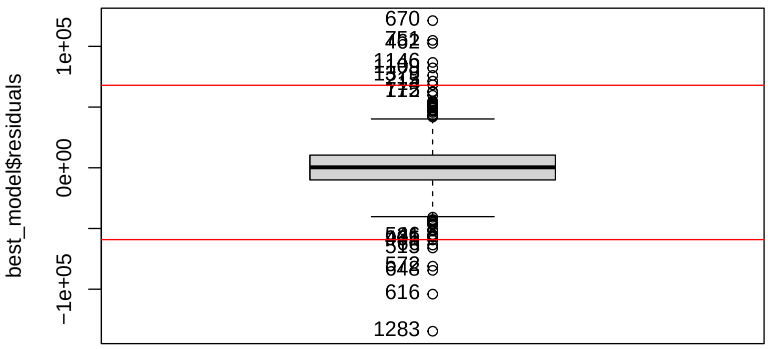
```
residuals_outliers
```

```
##  [1]   14   66  215  401  462  515  572  616  648  670  751  935 1109 1146 1283
## [16] 1379
```

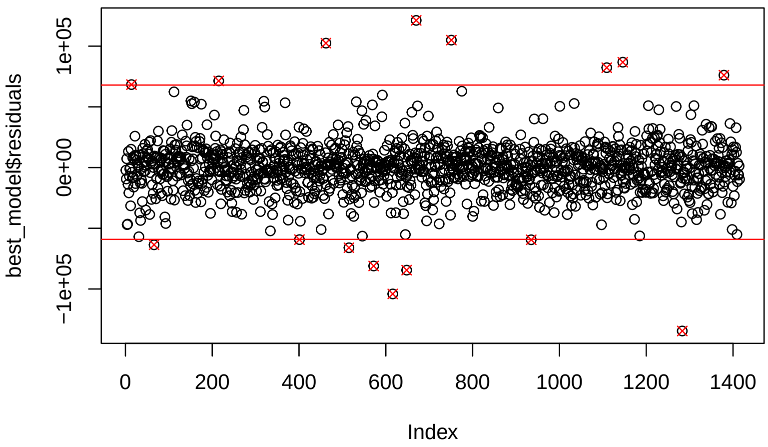
```
Boxplot(best_model$residuals)
```

```
##  [1] 1283  616  648  572  515   66  935  401   31  546  670  751  462 1146 1109
## [16] 1379  215   14  775  112
```

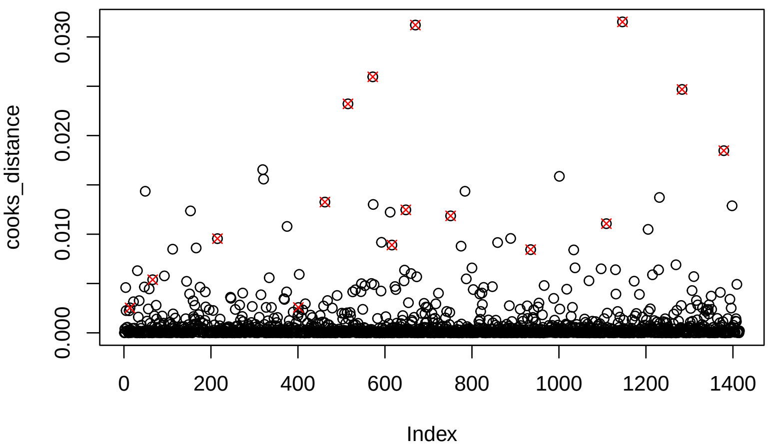
```
abline(h = residuals_upper_bound, col = "red")
abline(h = residuals_lower_bound, col = "red")
```




```r
plot(best_model$residuals)
abline(h = residuals_upper_bound, col = "red")
abline(h = residuals_lower_bound, col = "red")
points(
  residuals_outliers,
  best_model$residuals[residuals_outliers],
  pch = 4,
  col = "red"
)
```




```r
cooks_distance <- cooks.distance(best_model)
plot(cooks_distance)
points(residuals_outliers, cooks_distance[residuals_outliers], pch = 4, col = "red")
```




```r
residuals_outliers_df <- house_prices[residuals_outliers, ]
residuals_outliers_df$orig_idx <- residuals_outliers

library(broom)
res <- augment(m9)
res_outliers <- res[res$.rownames %in% residuals_outliers, ]
res_outliers <- res_outliers[order(abs(res_outliers$.resid), decreasing = TRUE), ]
res_outliers <- res_outliers[, c(".rownames", ".fitted", ".resid", "SalePrice", "Neighborhood", "OverallQual", "
print(res_outliers)
```

```
## # A tibble: 15 x 9
##    .rownames .fitted   .resid SalePrice Neighborhood         OverallQual LotArea
##    <chr>        <dbl>    <dbl>     <int> <fct>                <fct>         <int>
##  1 14         211162.  68338.    279500 College Creek        Good          10652
##  2 515        118728. -22228.     96500 Crawford             Average       10594
```

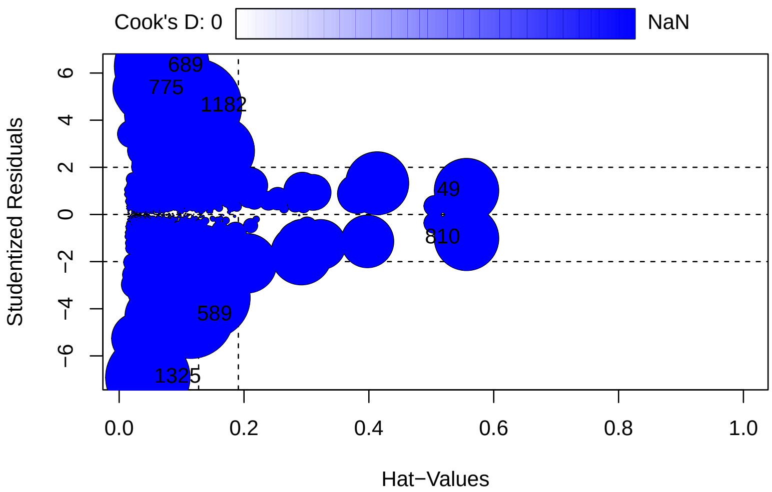
```
##  3 1379      100350. -17350.      83000 Briardale          Above Aver~    1953
##  4 1146      133652.  15348.     149000 Brookside          Average        6240
##  5 648       140062.  14938.     155000 Edwards            Above Aver~   10452
##  6 215       148546.  13204.     161750 College Creek      Above Aver~   10900
##  7 670       127571.   9929.     137500 Crawford           Below Aver~   11600
##  8 751        88693.   7807.      96500 Old Town           Below Aver~    8800
##  9 1109      175959.   5041.     181000 Gilbert            Above Aver~    8063
## 10 1283      155220.  -4720.     150500 College Creek      Average        8800
## 11 401       248515.  -3015.     245500 Veenker            Very Good     14963
## 12 462       153544.   1456.     155000 South & West of Iow~ Good         7200
## 13 572       121389.  -1389.     120000 North Ames         Above Aver~    7332
## 14 616       138755.  -1255.     137500 North Ames         Above Aver~    8800
## 15 66        316987.     13.1    317000 Northridge Heights Very Good      9591
## # i 2 more variables: GarageArea <int>, BedroomAbvGr <fct>
```

7 A priori influential data observations

The `influencePlot` function is used to create a plot of studentized residuals vs. hat values, identifying the observations with high leverage or high residuals.

8 a priori influential values were found

```
high_leverage <- as.data.frame(influencePlot(
  best_model,
  id = list(n = 3, method = "noteworthy")
))
```




```
mean_hat <- mean(high_leverage$Hat)
priori_influential <- row.names(high_leverage[
  which(high_leverage$hat > 3 * mean_hat)
])

priori_influential
```

```
## [1] "49"   "589"  "689"  "775"  "810"  "955"  "1182" "1325"
```

8 A posteriori influential data observations

The `dfbetas` function calculates the standardized difference in each parameter estimate with and without each observation, and it can be used to assess the effect of an individual observation on each estimated parameter of the fitted model. A large `dfbeta` value indicates that the observation has a large influence on the corresponding parameter estimate.

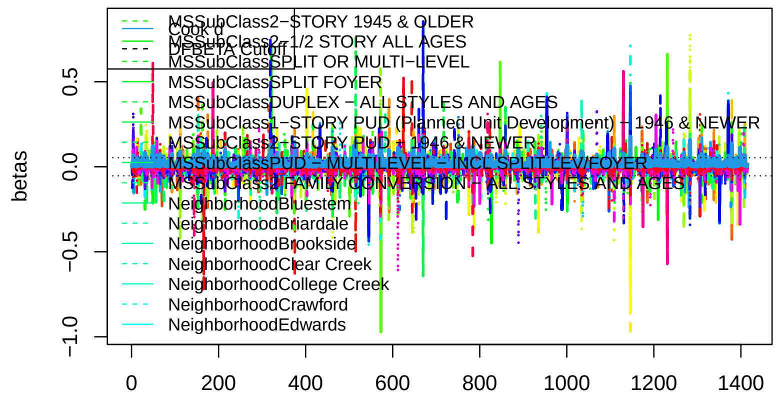
A dfbeta value greater than `2 / sqrt(dim(house_prices)[1])`, indicates a large influence on the parameter estimate. Those values are temporarily removed from the dataset and a new model is reconstructed with it. This new model demontrates an improvement in the R-squared value from 0.916 to 0.9773.

```
betas <- as.data.frame(dfbetas(best_model))
betas_cutoff <- 2 / sqrt(dim(house_prices)[1])
betas_cutoff
```

```
## [1] 0.05316818
```

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

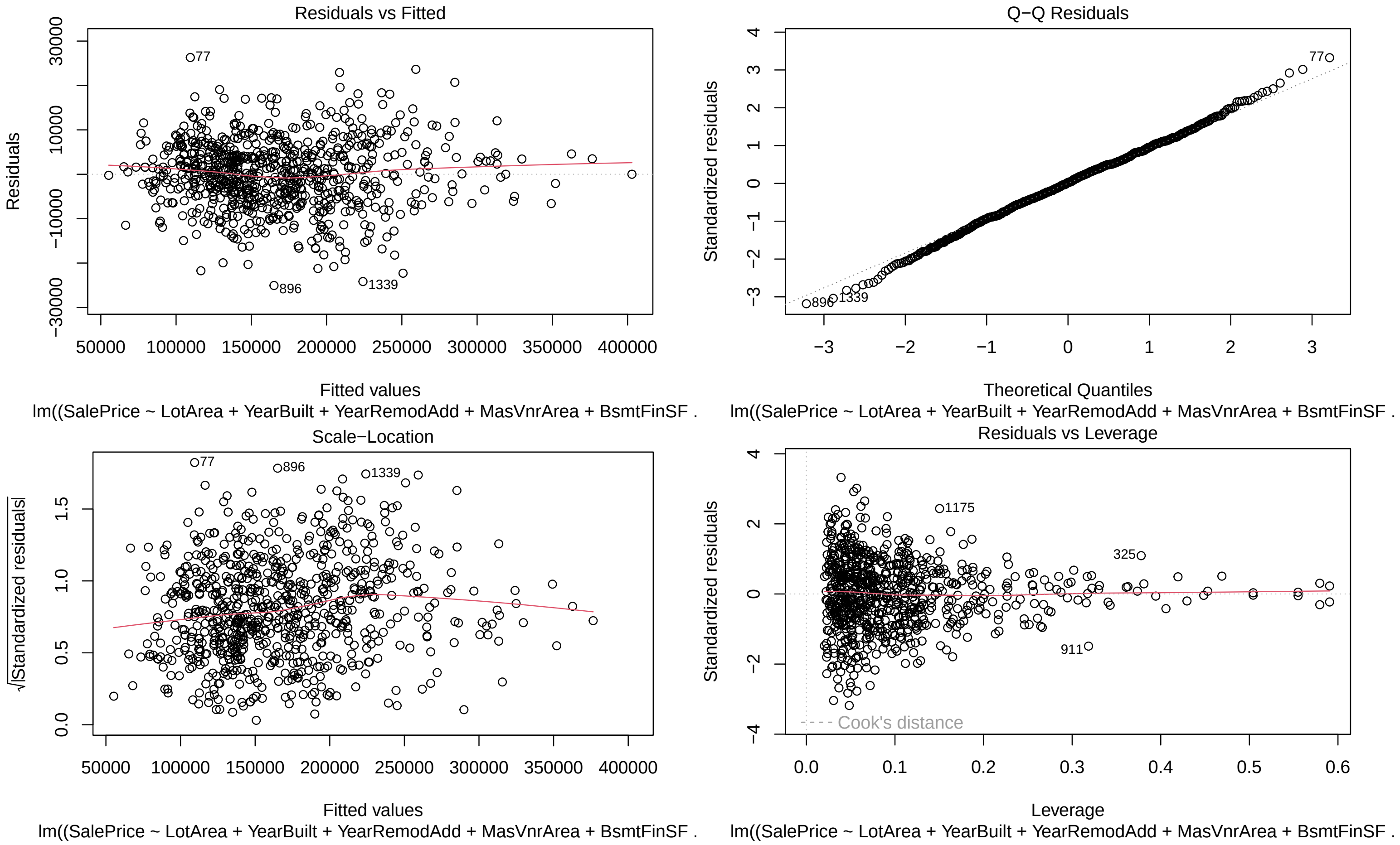
```r
  betas,
  type = "l",
  lwd = 2,
  col = rainbow(ncol(betas))
)
lines(
  sqrt(cooks.distance(best_model)),
  col = 4,
  lwd = 3
)
abline(
  h = betas_cutoff,
  lty = 3,
  lwd = 1,
  col = 1
)
abline(
  h = -betas_cutoff[1],
  lty = 3,
  lwd = 1,
  col = 1
)
legend(
  "topleft",
  legend = c("Cook d", "DFBETA Cutoff"),
  col = c(4, 1),
  lty = 1:2,
  cex = 0.8
)

legend(
  "bottomleft",
  legend = names(coef(best_model)),
  col = rainbow(ncol(betas)),
  lty = 1:2,
  cex = 0.8,
  ncol = 2
)
```




```r
large_df <- apply(betas, 1, function(x) any(abs(x) > betas_cutoff))
reduced_data <- house_prices[!large_df, ]
new_model <- lm(
  formula = (
    SalePrice ~ LotArea + YearBuilt + YearRemodAdd +
    MasVnrArea + BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF + X1stFlrSF +
    X2ndFlrSF + GarageArea + WoodDeckSF + OpenPorchSF + ScreenPorch +
    ExterQual + BsmtQual + KitchenQual + MSSubClass + Neighborhood +
    f.YearBuilt + OverallQual + HalfBath + BedroomAbvGr + KitchenAbvGr +
    Fireplaces
  ),
  data = reduced_data
)

# summary(new_model)
# summary(best_model)
```

```r
# par(mfrow = c(2, 2))
# plot(best_model)

plot(new_model)
```

```
## Warning: not plotting observations with leverage one:
##   100, 274, 314, 322, 508
```




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

9 Model testing with test samples

9.1 Load and prepare Test Data

We prepared the test data by retaining only the variables that were used in the model. Upon analysis, we observed the emergence of new levels in MSSubClass, Neighborhood, OverallQual, BedroomAbvGr, and Fireplaces, which were not present in the training dataset. Given that these levels represented only a small number of values, we opted to either eliminate them or combine them with another level, as our model is not equipped to handle them. Additionally, we removed three instances with missing values from GarageArea, BsmtUnfSF, BsmtFinSF1, KitchenQual, BsmtFinSF2, and MasVnrArea, in an effort to potentially mitigate bias introduced by imputation. Finally, we performed imputation on the missing values in MasVnrArea, following the same approach as we did with the training dataset.

```r
test_data <- read.csv("test.csv")
na_factor_cols <- c("BsmtQual", "GarageFinish", "FireplaceQu", "GarageType")

test_data[na_factor_cols] <- lapply(
  test_data[na_factor_cols],
  function(x) {
    replace_na(x, "NA")
  }
)
#Prepare Test Data
selected_variables <- c("LotArea", "YearBuilt", "YearRemodAdd", "MasVnrArea",
                        "BsmtFinSF1", "BsmtFinSF2", "BsmtUnfSF", "X1stFlrSF",
                        "X2ndFlrSF", "GarageArea", "WoodDeckSF", "OpenPorchSF",
                        "ScreenPorch", "ExterQual", "BsmtQual", "KitchenQual",
                        "MSSubClass", "Neighborhood", "OverallQual",
                        "HalfBath", "BedroomAbvGr", "KitchenAbvGr", "Fireplaces")
```

```r
test_data <- test_data[selected_variables]
# Specify the variables to be converted to factors
factor_variables <- c("ExterQual", "BsmtQual", "KitchenQual",
                      "MSSubClass", "Neighborhood", "OverallQual",
                      "HalfBath", "BedroomAbvGr", "KitchenAbvGr", "Fireplaces")
# Convert specified variables to factors in test_data
test_data[factor_variables] <- lapply(
  test_data[factor_variables],
  function(var) factor(var)
)

test_data$f.YearBuilt <- factor(
  cut(
    test_data$YearBuilt,
    breaks = c(1872, 1915, 1945, 1960, 1980, 2000, 2010),
    labels = c("Historic", "Pre-War", "Post-War", "Mid-Century", "Modern", "Contemporary"),
    right = TRUE,
    include.lowest = TRUE
  )
)

cols <- c(
  "OverallQual", "Neighborhood", "ExterQual", "BsmtQual", "KitchenQual", "MSSubClass"
)
levels_list <- list(
  1:10, # OverallQual
  c(
    "Blmngtn", "Blueste", "BrDale", "BrkSide", "ClearCr", "CollgCr", "Crawfor",
    "Edwards", "Gilbert", "IDOTRR", "MeadowV", "Mitchel", "NAmes", "NoRidge",
    "NPkVill", "NridgHt", "NWAmes", "OldTown", "SWISU", "Sawyer", "SawyerW",
    "Somerst", "StoneBr", "Timber", "Veenker"
  ), # Neighborhood
  c("Ex", "Gd", "TA", "Fa", "Po"), # ExterQual
  c("Ex", "Gd", "TA", "Fa", "Po", "NA"), # BsmtQual
  c("Ex", "Gd", "TA", "Fa", "Po"), # KitchenQual
  c(
    "20", "30", "40", "45", "50", "60", "70", "75", "80", "85", "90", "120",
    "150", "160", "180", "190"
  ) # MSSubClass
)

labels_list <- list(
  c(
    "Very Poor", "Poor", "Fair", "Below Average", "Average", "Above Average",
    "Good", "Very Good", "Excellent", "Very Excellent"
  ), # OverallQual
  c(
    "Bloomington Heights", "Bluestem", "Briardale", "Brookside", "Clear Creek",
    "College Creek", "Crawford", "Edwards", "Gilbert", "Iowa DOT and Rail Road",
    "Meadow Village", "Mitchell", "North Ames", "Northridge", "Northpark Villa",
    "Northridge Heights", "Northwest Ames", "Old Town",
    "South & West of Iowa State University", "Sawyer", "Sawyer West",
    "Somerset", "Stone Brook", "Timberland", "Veenker"
  ), # Neighborhood
  c("Excellent", "Good", "Average/Typical", "Fair", "Poor"), # ExterQual
  c(
    "Excellent (100+ inches)", "Good (90-99 inches)", "Typical (80-89 inches)",
    "Fair (70-79 inches)", "Poor (<70 inches)", "No Basement"
  ), # BsmtQual
  c("Excellent", "Good", "Typical/Average", "Fair", "Poor"), # KitchenQual
  c(
    "1-STORY 1946 & NEWER ALL STYLES", "1-STORY 1945 & OLDER",
    "1-STORY W/FINISHED ATTIC ALL AGES", "1-1/2 STORY - UNFINISHED ALL AGES",
    "1-1/2 STORY FINISHED ALL AGES", "2-STORY 1946 & NEWER",
    "2-STORY 1945 & OLDER", "2-1/2 STORY ALL AGES", "SPLIT OR MULTI-LEVEL",
    "SPLIT FOYER",
    "DUPLEX - ALL STYLES AND AGES",
```

```r
    "1-STORY PUD (Planned Unit Development) - 1946 & NEWER",
    "1-1/2 STORY PUD - ALL AGES", "2-STORY PUD - 1946 & NEWER",
    "PUD - MULTILEVEL - INCL SPLIT LEV/FOYER",
    "2 FAMILY CONVERSION - ALL STYLES AND AGES"
  ) # MSSubClass
)

test_data[cols] <- lapply(
  seq_along(cols),
  function(i) {
    factor(
      test_data[[cols[i]]],
      levels = levels_list[[i]],
      labels = labels_list[[i]]
    )
  }
)

#New factors deletion
#MSSubClass
# table(test_data$MSSubClass)
prop.table(table(test_data$MSSubClass))
```

```
##
##                      1-STORY 1946 & NEWER ALL STYLES
##                                          0.372172721
##                                 1-STORY 1945 & OLDER
##                                          0.047978067
##                    1-STORY W/FINISHED ATTIC ALL AGES
##                                          0.001370802
##                    1-1/2 STORY - UNFINISHED ALL AGES
##                                          0.004112406
##                        1-1/2 STORY FINISHED ALL AGES
##                                          0.098012337
##                                 2-STORY 1946 & NEWER
##                                          0.189170665
##                                 2-STORY 1945 & OLDER
##                                          0.046607265
##                                 2-1/2 STORY ALL AGES
##                                          0.004797807
##                                 SPLIT OR MULTI-LEVEL
##                                          0.041124058
##                                          SPLIT FOYER
##                                          0.019191227
##                           DUPLEX - ALL STYLES AND AGES
##                                          0.039067855
## 1-STORY PUD (Planned Unit Development) - 1946 & NEWER
##                                          0.065113091
##                              1-1/2 STORY PUD - ALL AGES
##                                          0.000685401
##                              2-STORY PUD - 1946 & NEWER
##                                          0.044551062
##                 PUD - MULTILEVEL - INCL SPLIT LEV/FOYER
##                                          0.004797807
##               2 FAMILY CONVERSION - ALL STYLES AND AGES
##                                          0.021247430
```

```r
# Create a logical condition for filtering
condition <- !(test_data$MSSubClass %in% c("1-STORY W/FINISHED ATTIC ALL AGES", "1-1/2 STORY PUD - ALL AGES"))
# Subset test_data based on the condition
test_data <- test_data[condition, ]
#Neighborhood
# table(test_data$Neighborhood)
prop.table(table(test_data$Neighborhood))
```

```
##
##                  Bloomington Heights                                 Bluestem
##                          0.007554945                              0.005494505
```

```
##                          Briardale                          Brookside
##                        0.009615385                        0.033653846
##                        Clear Creek                      College Creek
##                        0.010302198                        0.080357143
##                           Crawford                           Edwards
##                        0.035714286                        0.063873626
##                            Gilbert           Iowa DOT and Rail Road
##                        0.059065934                        0.038461538
##                     Meadow Village                          Mitchell
##                        0.013736264                        0.044642857
##                         North Ames                        Northridge
##                        0.149725275                        0.020604396
##                    Northpark Villa                Northridge Heights
##                        0.009615385                        0.061126374
##                     Northwest Ames                          Old Town
##                        0.039835165                        0.086538462
## South & West of Iowa State University                          Sawyer
##                        0.015796703                        0.052884615
##                        Sawyer West                          Somerset
##                        0.045329670                        0.065934066
##                        Stone Brook                        Timberland
##                        0.017857143                        0.023351648
##                            Veenker
##                        0.008928571
```

```r
ll <- which(test_data$Neighborhood == "Bluestem");ll
```

```
## [1]  139  140  448  449  450  750 1108 1110
```

```r
test_data <- test_data[-ll, ]
#OverallQual
# table(test_data$OverallQual)
prop.table(table(test_data$OverallQual))
```

```
##
##      Very Poor           Poor           Fair  Below Average        Average
##    0.001381215    0.006906077    0.013812155    0.075966851    0.294889503
##  Above Average           Good      Very Good      Excellent Very Excellent
##    0.242403315    0.193370166    0.118093923    0.044198895    0.008977901
```

```r
ll <- which(test_data$OverallQual == "Poor" | test_data$OverallQual == "Very Poor");ll
```

```
##  [1]   77  139  326  353  386  451  634  641  751 1109 1401 1434
```

```r
test_data <- test_data[-ll, ]
#BedroomAbvGr
# table(test_data$BedroomAbvGr)
prop.table(table(test_data$BedroomAbvGr))
```

```
##
##           0           1           2           3           4           5
## 0.001392758 0.030640669 0.258356546 0.550835655 0.130222841 0.018802228
##           6
## 0.009749304
```

```r
ll <- which(test_data$BedroomAbvGr == "0");ll
```

```
## [1] 1038 1121
```

```r
test_data <- test_data[-ll, ]
test_data$BedroomAbvGr <- replace(test_data$BedroomAbvGr, test_data$BedroomAbvGr == 6, 5)
#Fireplaces
# table(test_data$Fireplaces)
prop.table(table(test_data$Fireplaces))
```

```
##
##            0            1            2            3            4
## 0.4993026499 0.4239888424 0.0718270572 0.0041841004 0.0006973501
```

```r
ll <- which(test_data$Fireplaces == "4");ll
```

```
## [1] 1229
```

```r
test_data <- test_data[-ll, ]

# summary(test_data)
# Missing values
ll_na <- which(is.na(test_data$GarageArea) | is.na(test_data$BsmtUnfSF) | is.na(test_data$BsmtFinSF1) | is.na(te
```

```
## [1]   95  648 1097
```

```r
#Discard observations with NA's
test_data <- test_data[-ll_na,]
#Impute MasVnrArea
res.pca <- imputePCA(test_data[, c(1:13)])
# summary(res.pca$completeObs)
test_data$MasVnrArea <- res.pca$completeObs[, 4]
# summary(test_data)
```

9.2 Make predictions

Observing the test dataset, we noticed that the SalePrice variable was not provided. Consequently, we were unable to calculate the accuracy of our prediction. Nevertheless, the interactions between the categorical and numerical variables with the predicted variable closely resemble those in the training dataset. Furthermore, for having a test dataset with the actual target variable, we decided to divide our train dataset into two datasets with the **caret** package. We trained our model again with this new split data, and then validated the model with the new test dataset. The interpretations of the obtained results are the following ones. - The Coefficient of Variation (CV) of the test dataset (0.3889199) is relatively small compared to the RMSE ratio (0.1214806). This can be seen as a positive aspect, indicating that the model's errors are relatively small relative to the average size of the response variable. - The R-squared of 0.9022038 indicates that approximately 90.22% of the variability in the SalePrice can be explained by the independent variables included in the model. - Looking at the Scatter Plot, we can say that the points are close to the (y=x) diagonal. Nevertheless their are still some large residuals, because of outliers and missing interactions.

```r
final_model <- new_model
predictions <- predict(final_model, newdata = test_data)
test_data$PredictedSalePrice <- predictions
res.con <- condes(test_data, num.var = 25)
res.con$quanti
```

```
##              correlation       p.value
## GarageArea    0.69177661 3.725203e-204
## X1stFlrSF     0.67808555 3.442113e-193
## YearBuilt     0.62468268 1.373153e-155
## YearRemodAdd  0.57492204 1.360871e-126
## MasVnrArea    0.56479150 2.778517e-121
## BsmtFinSF1    0.51161937  3.236878e-96
## LotArea       0.36922609  2.034278e-47
## OpenPorchSF   0.36152816  2.140828e-45
## WoodDeckSF    0.35065379  1.241689e-42
## X2ndFlrSF     0.22311198  1.371380e-17
## BsmtUnfSF     0.15450598  4.282741e-09
## ScreenPorch   0.09187979  5.036877e-04
```

```r
res.con$quali
```

```
##                      R2       p.value
## Neighborhood 0.66137666  0.000000e+00
## OverallQual  0.78511824  0.000000e+00
## BsmtQual     0.61240754 2.281581e-291
## ExterQual    0.57839923 8.227372e-267
## KitchenQual  0.54086303 2.045793e-240
## f.YearBuilt  0.45676665 9.061257e-186
## Fireplaces   0.26888260  1.650633e-96
## MSSubClass   0.28629275  3.467698e-94
## HalfBath     0.09316101  4.986386e-31
## BedroomAbvGr 0.02965136  1.073926e-08
## KitchenAbvGr 0.01566311  1.282552e-05
```

```r
library(caret)
```

```
## Loading required package: lattice
```

```r
# Set seed for reproducibility
set.seed(123)
```

```r
# Create an index for splitting the data
index <- createDataPartition(house_prices$SalePrice, p = 0.7, list = FALSE)
# Create training and testing datasets
train_data <- house_prices[index, ]
test_data <- house_prices[-index, ]

final_model2 <- lm(
  formula = SalePrice ~ LotArea + YearBuilt + YearRemodAdd +
            MasVnrArea + BsmtFinSF1 + BsmtFinSF2 + BsmtUnfSF + X1stFlrSF +
            X2ndFlrSF + GarageArea + WoodDeckSF + OpenPorchSF + ScreenPorch +
            ExterQual + BsmtQual + KitchenQual + MSSubClass + Neighborhood +
            f.YearBuilt + OverallQual + HalfBath + BedroomAbvGr + KitchenAbvGr +
            Fireplaces,
  data = train_data
)
# summary(final_model2)
#Validate
predictions <- predict(final_model2, newdata = test_data)
actual_values <- test_data$SalePrice

rmse <- sqrt(mean((predictions - actual_values)^2))
r_squared <- 1 - sum((actual_values - predictions)^2) / sum((actual_values - mean(actual_values))^2)

cat("Root Mean Squared Error (RMSE):", rmse, "\n")
```

```
## Root Mean Squared Error (RMSE): 21392.4
```

```r
cv_response_variable <- sd(test_data$SalePrice) / mean(test_data$SalePrice);cv_response_variable
```

```
## [1] 0.3889199
```

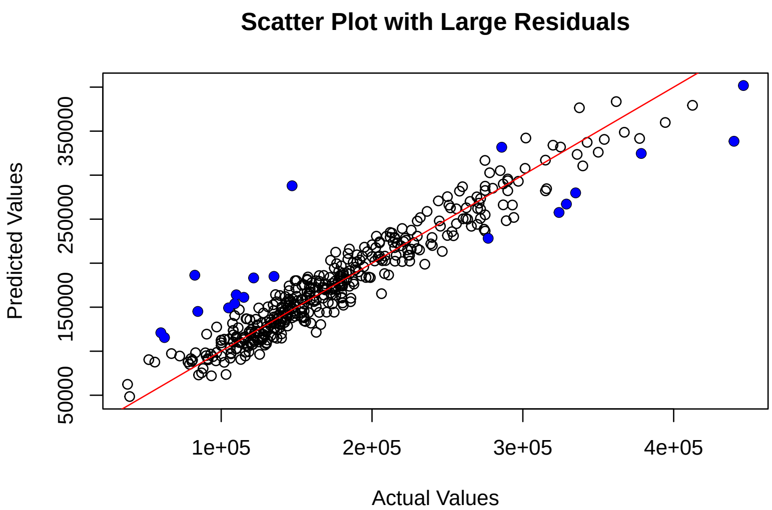
```r
cv_rmse_ratio <- rmse / mean(test_data$SalePrice);cv_rmse_ratio
```

```
## [1] 0.1214806
```

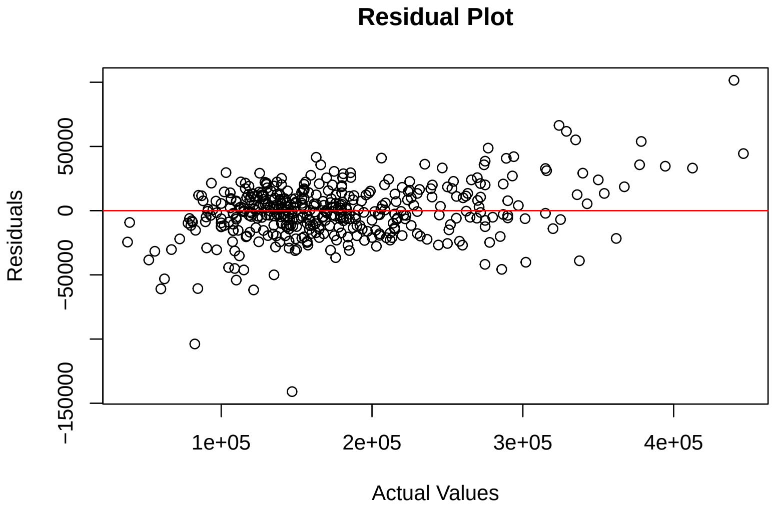
```r
cat("R-squared:", r_squared, "\n")
```

```
## R-squared: 0.9022038
```

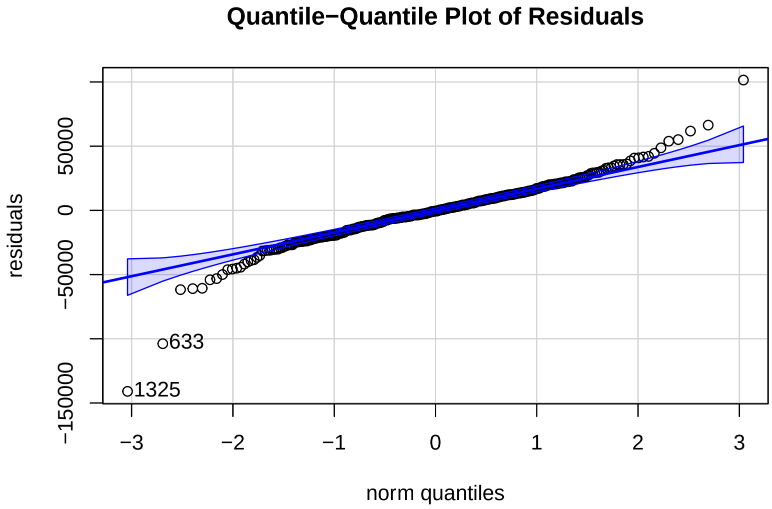
```r
residuals <- actual_values - predictions

large_residual_threshold <- 2 * sd(residuals)
# Identify indices of points with large residuals
large_residual_indices <- which(abs(residuals) > large_residual_threshold)
observations_with_large_residuals <- test_data[large_residual_indices, ]
plot(actual_values, predictions, main = "Scatter Plot with Large Residuals", xlab = "Actual Values", ylab = "Pre
abline(0, 1, col = "red")  # Add a diagonal line for reference
points(actual_values[large_residual_indices], predictions[large_residual_indices], col = "blue", pch = 16)
```




```r
plot(actual_values, residuals, main = "Residual Plot", xlab = "Actual Values", ylab = "Residuals")
abline(h = 0, col = "red")  # Add a horizontal line at y = 0 for reference
```




```r
qqPlot(residuals, main = "Quantile-Quantile Plot of Residuals")
```




```
## 1325  633
##  386  170
```