House Price Prediction with R Part I

Md Mominul Islam

11/12/2021

House Price Prediction

Prediction in research satisfies one of humankind's greatest primal desires. We make a lot of predictions to see into the future and know what awaits us. Most scientists regularly use prediction in research as a fundamental of the scientific method, when they generate a hypothesis and predict what will happen. Machine learning models function similarly to humans. They use instances/logics they've seen before to forecast what will happen given a certain set of circumstances.

For this lesson, we will bulid a model to figure out how much a house will sell for?

The steps that we will follow are given below:

1. Load Package
2. Read Train and Test data
3. Combine Train and Test dataset
4. Data Visualization
5. Missing Value
6. Removing Skewed Variables
7. Build the model
8. Variable importance
9. Final Prediction
10. Calculate RMSE

Loading and Exploring Data

Loading libraries required and reading the data into R

Loading R packages used besides base R.

```
library(knitr)
library(ggplot2)
library(plyr)
library(dplyr)
library(corrplot)
library(caret)
```

```
library(gridExtra)
library(scales)
library(Rmisc)
library(ggrepel)
library(randomForest)
library(psych)
library(xgboost)
```

The first thing we want to do is familiarize ourselves with the data. By checking the dataset, we can have a holistic picture about the variables that are present in the dataset.

```
train.data <- read.csv(("Data Set/train.csv"), stringsAsFactors = F)
test.data <- read.csv(("Data Set/test.csv"), stringsAsFactors = F)
```

Structure of the data

```
dim(train.data)
str(train.data)
dim(test.data)
str(test.data)
```

The housing train data set has 1460 rows and 81 features with the target feature "SalePrice". The housing test data set has 1459 rows and 80 features with the target feature Sale Price.

```
#Count the number of columns that consists of text data
sum(sapply(train.data[,1:81], typeof) == "character")
```

```
## [1] 43
```

```
#Count the number of columns that consists of numerical data
sum(sapply(train.data[,1:81], typeof) == "integer")
```

```
## [1] 38
```

We have 43 columns that consist of character variable and 38 columns are numerical. The text data could be challenging to work with. For those that are numerical,let's take a look at some descriptive statistics.

```
cat('Train data has', dim(train.data)[1], 'rows and', dim(train.data)[2], 'columns.')
```

```
## Train data has 1460 rows and 81 columns.
```

```
cat('Test data has', dim(test.data)[1], 'rows and', dim(test.data)[2], ' columns.')
```

```
## Test data has 1459 rows and 80  columns.
```

```r
# The percentage of data missing in train
tr.miss <- (sum(is.na(train.data)) / (nrow(train.data) *ncol(train.data)))*100
cat("Missing value Percentage for Train Data is", tr.miss, '%')
```

```
## Missing value Percentage for Train Data is 5.889565 %
```

```r
# The percentage of data missing in test
tst.miss <- (sum(is.na(test.data)) / (nrow(test.data) * ncol(test.data)))*100
cat("Missing value Percentage for Train Data is", tst.miss, '%')
```

```
## Missing value Percentage for Train Data is 5.997258 %
```

```r
# Check for duplicated rows
cat("The number of duplicated rows are", nrow(train.data) - nrow(unique(train.data)))
```

```
## The number of duplicated rows are 0
```

We have 5.88% missing values in our train data set and 7.15% missing values in our test data set.

Combine data

Since test data set has no "Saleprice" variable. We will create it and then combine.

```r
test.data$SalePrice<-rep(NA,1459)
combined_data <-bind_rows(train.data,test.data)
all <- rbind(train.data,test.data)
dim(all)
```

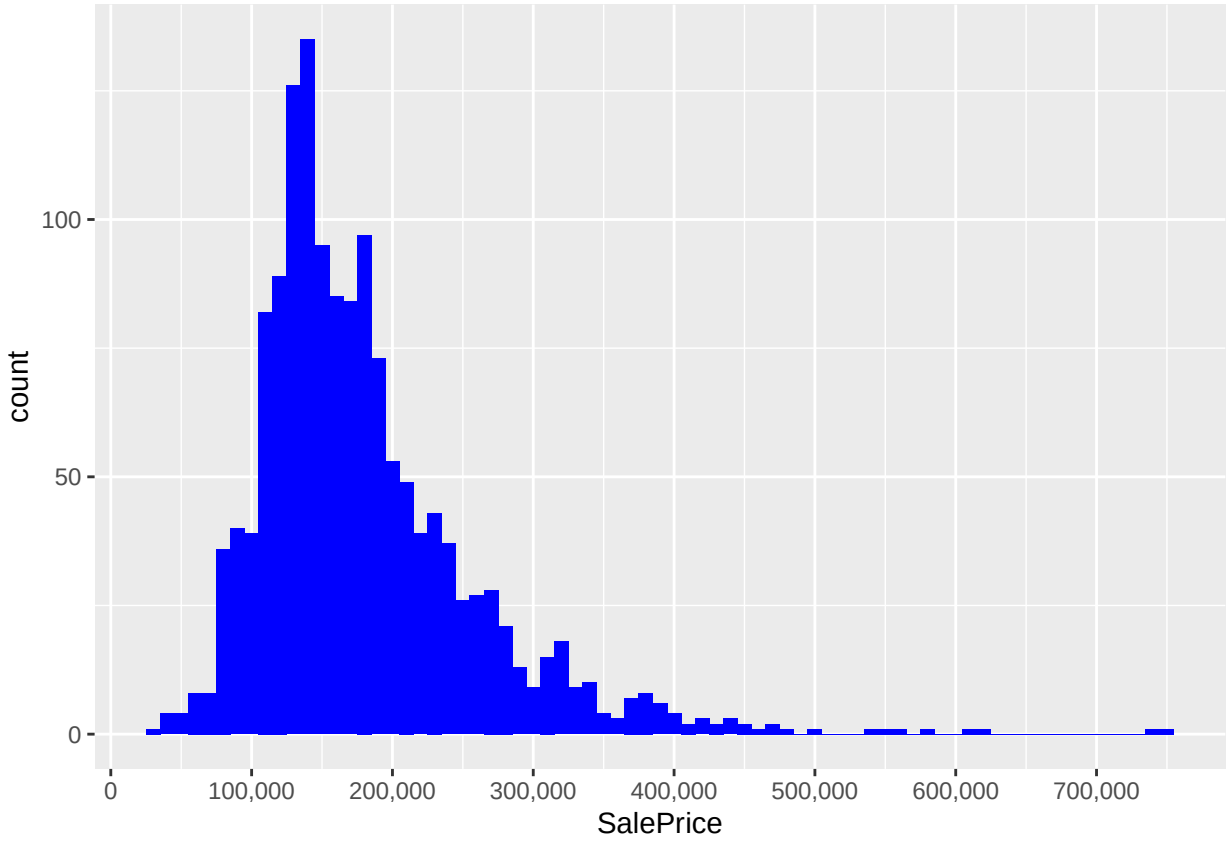
```
## [1] 2919   81
```

After combining the data set, we can clearly see that there are 81 columns in our data set.

Data Exploration

The response variable: SalePrice

```r
ggplot(data=all[!is.na(all$SalePrice),], aes(x=SalePrice)) +
        geom_histogram(fill="blue", binwidth = 10000) +
        scale_x_continuous(breaks= seq(0, 800000, by=100000), labels = comma)
```



As we can see, the sale prices are right skewed. This was expected as few people can afford very expensive houses. We will keep this in mind, and take measures before modeling.

Summary Statistics for SalePrice

```
kable(as.array(summary(all$SalePrice)),
      caption = 'Summary Statistics of Sales Price')
```

Table 1: Summary Statistics of Sales Price

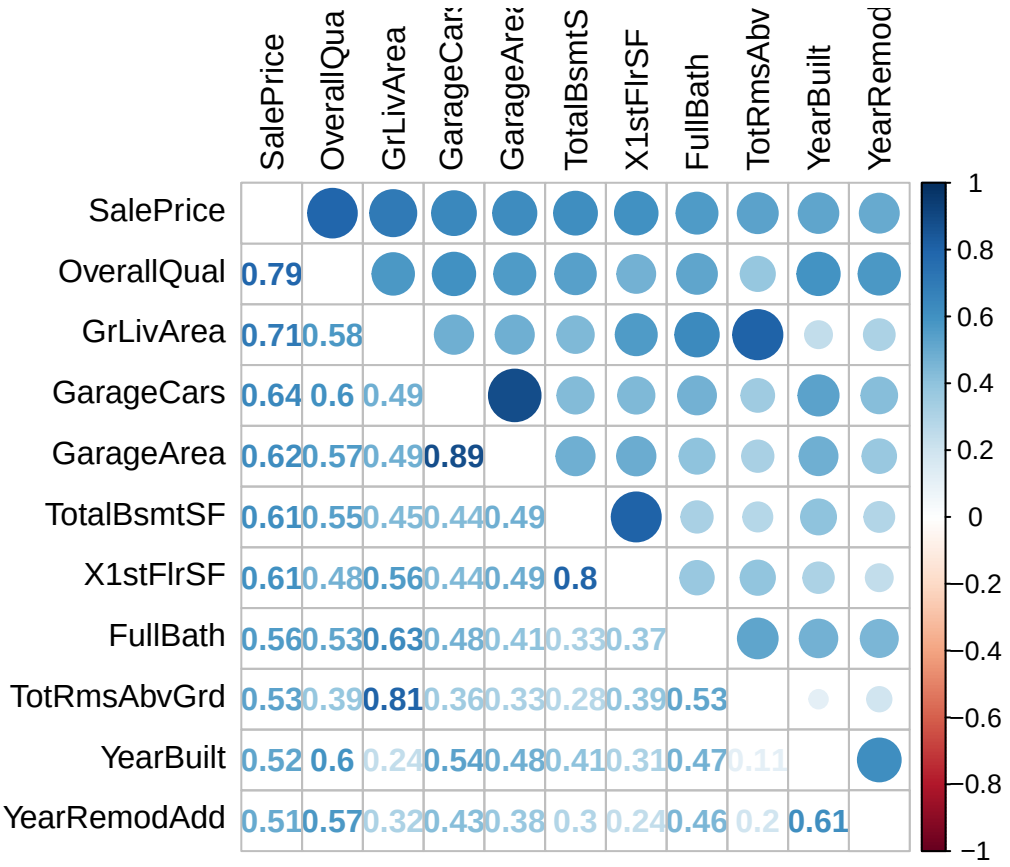
Var1	Freq
Min.	34900.0
1st Qu.	129975.0
Median	163000.0
Mean	180921.2
3rd Qu.	214000.0
Max.	755000.0
NA's	1459.0

Important Numeric Predictors

The character variables need some work before we can use them. To get a feel for the dataset, I decided to first see which numeric variables have a high correlation with the SalePrice.

Correlations with SalePrice

```
numericVars <- which(sapply(all, is.numeric)) #index vector numeric variables
numericVarNames <- names(numericVars) #saving names vector for use later on
all_numVar <- all[, numericVars]
cor_numVar <- cor(all_numVar, use="pairwise.complete.obs") #correlations of all numeric variables
#sort on decreasing correlations with SalePrice
cor_sorted <- as.matrix(sort(cor_numVar[,'SalePrice'], decreasing = TRUE))
 #select only high corelations
CorHigh <- names(which(apply(cor_sorted, 1, function(x) abs(x)>0.5)))
cor_numVar <- cor_numVar[CorHigh, CorHigh]
corrplot.mixed(cor_numVar, tl.col="black", tl.pos = "lt")
```



We have 38 numerical variables in our data set.Altogether, there are 10 numeric variables with a correlation of at least 0.5 with SalePrice. All those correlations are positive. As from the correlation matrix plot, we can see that OverallQual, GrLivArea, GarageCars are highly correlated with the SalePrice. We have to keep in mind about multicollinearity. For example: the correlation between GarageCars and GarageArea is very high (0.89), and both have similar (high) correlations with SalePrice.

The other 6 six variables with a correlation higher than 0.5 with SalePrice are:

-TotalBsmtSF: Total square feet of basement area

-1stFlrSF: First Floor square feet

-FullBath: Full bathrooms above grade

-TotRmsAbvGrd: Total rooms above grade (does not include bathrooms)

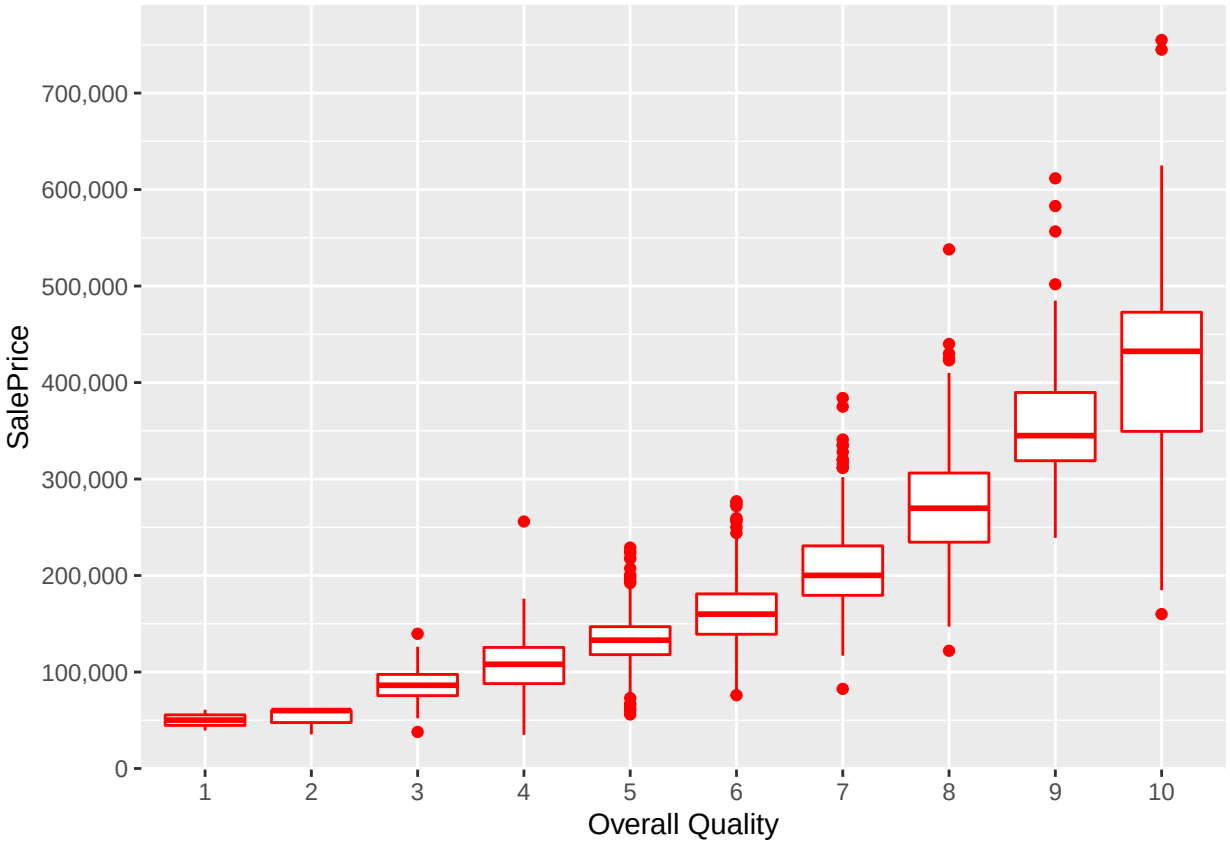
-YearBuilt: Original construction date

-YearRemodAdd: Remodel date (same as construction date if no remodeling or additions)

Overall Quality

Overall Quality has the highest correlation with SalePrice among the numeric variables (0.79). It rates the overall material and finish of the house on a scale from 1 (very poor) to 10 (very excellent).

```
ggplot(data=all[!is.na(all$SalePrice),], aes(x=factor(OverallQual), y=SalePrice))+
        geom_boxplot(col='red') + labs(x='Overall Quality') +
        scale_y_continuous(breaks= seq(0, 800000, by=100000), labels = comma)
```



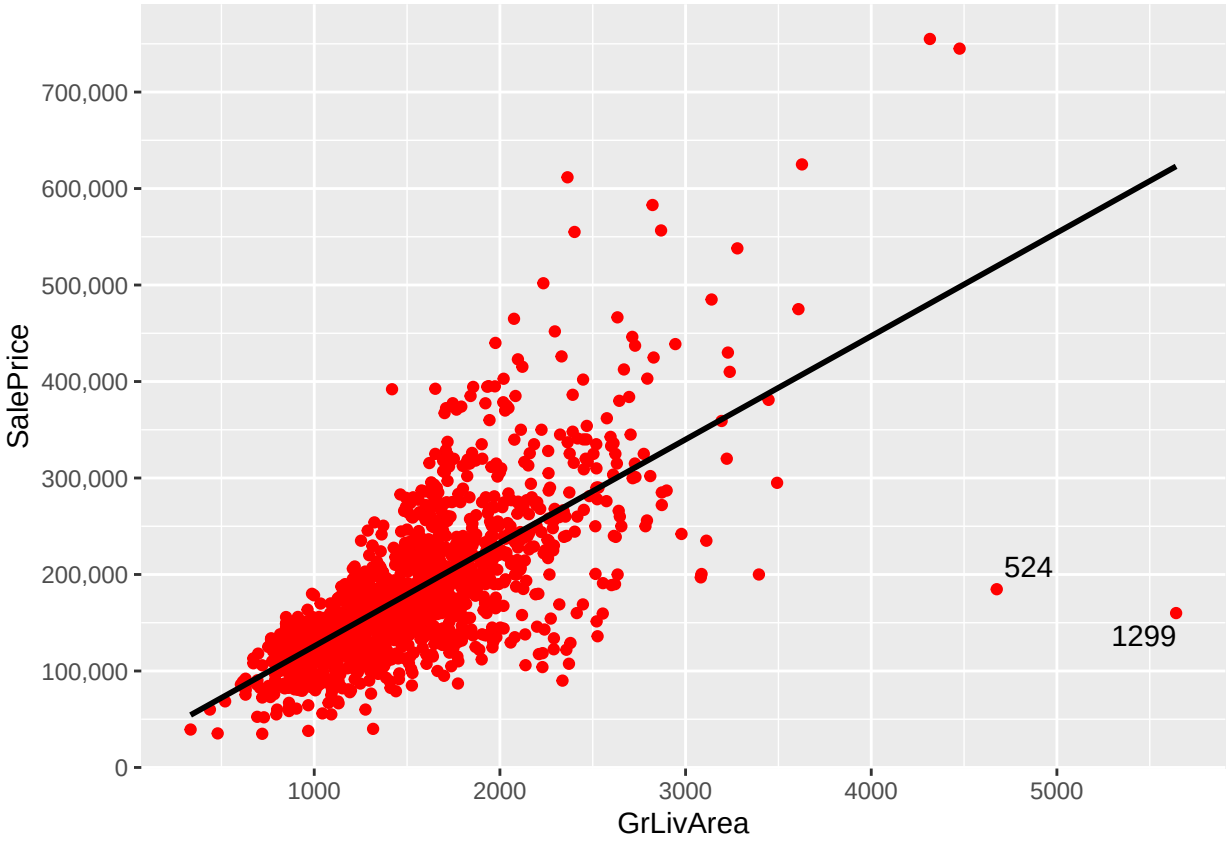
The positive correlation is certainly there indeed, and seems to be a slightly upward curve. Regarding outliers, we do not see any extreme values. If there is a candidate to take out as an outlier later on, it seems to be the expensive house with grade 4.

Above Grade (Ground) Living Area (square feet)

The numeric variable with the second highest correlation with SalesPrice is the Above Grade Living Area. This make a lot of sense; big houses are generally more expensive.

```
ggplot(data=all[!is.na(all$SalePrice),], aes(x=GrLivArea, y=SalePrice))+
        geom_point(col='red') + geom_smooth(method = "lm", se=FALSE, color="black", aes(group=1)) +
        scale_y_continuous(breaks= seq(0, 800000, by=100000), labels = comma) +
        geom_text_repel(aes(label = ifelse(all$GrLivArea[!is.na(all$SalePrice)]>4500, rownames(all), ''
```

```
## `geom_smooth()` using formula 'y ~ x'
```



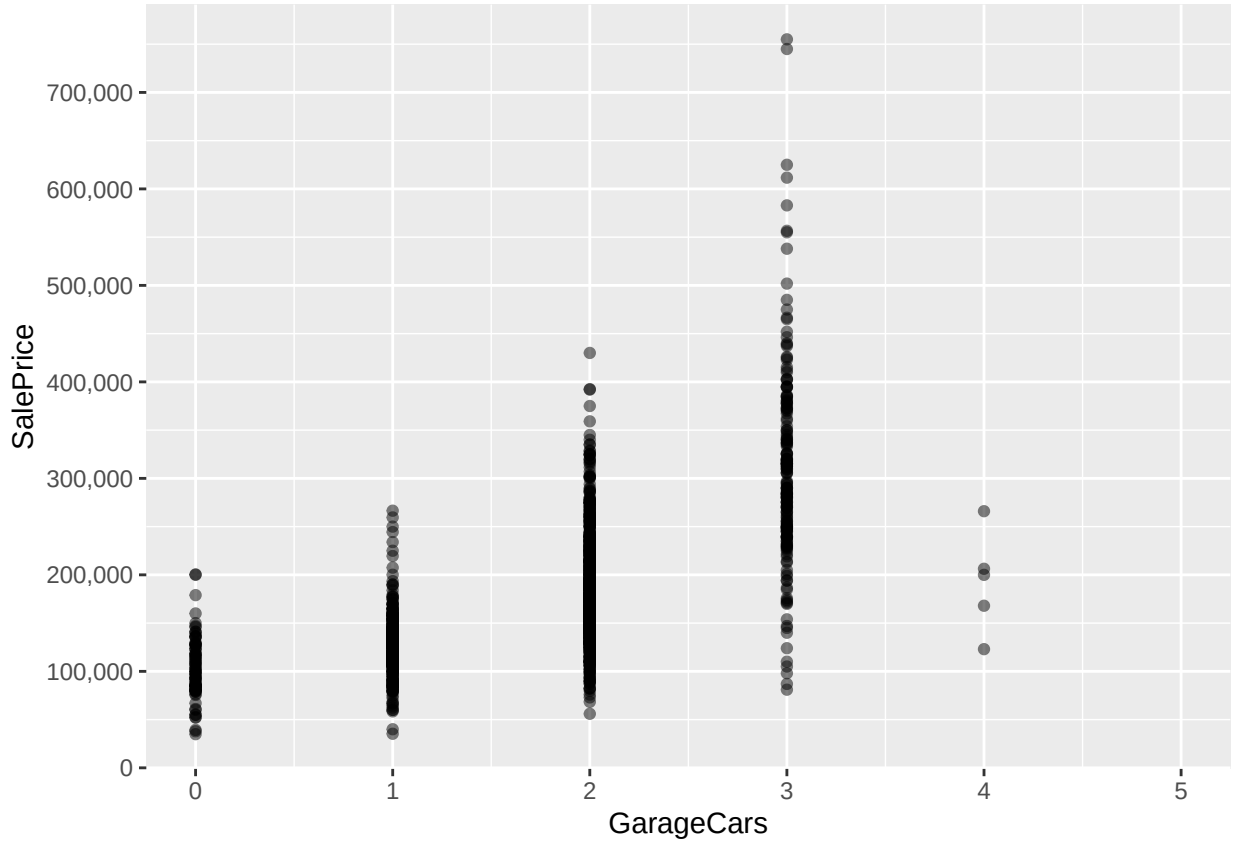
Especially the two houses with really big living areas and low SalePrices seem outliers (houses 524 and 1299, see labels in graph). we will not take them out yet, as taking outliers can be dangerous. For instance, a low score on the Overall Quality could explain a low price. However, as we can see below, these two houses actually also score maximum points on Overall Quality. Therefore, we will keep houses 1299 and 524 in mind as prime candidates to take out as outliers.

```
#Looking at highly correlated values of outliers
a.outlr <- all[c(524, 1299), c('SalePrice', 'GrLivArea', 'OverallQual')]
kable(a.outlr)
```

	SalePrice	GrLivArea	OverallQual
524	184750	4676	10
1299	160000	5642	10

Garage Cars

```
ggplot(all, aes(x = GarageCars, y = SalePrice)) +
  geom_point(alpha = .5) +
  geom_smooth()+
  scale_y_continuous(breaks= seq(0, 800000, by=100000), labels = comma)
```



As we can see, if there in no garage for cars then the price is low. With the increase of GarageCars, SalePrice increases.

Missing data, label encoding, and factorizing variables

Completeness of the data

First of all, we would like to see which variables contain missing values.

```
NAcol <- which(colSums(is.na(all)) > 0)
sort(colSums(sapply(all[NAcol], is.na)), decreasing = TRUE)
```

```
##       PoolQC  MiscFeature        Alley        Fence    SalePrice  FireplaceQu
##         2909         2814         2721         2348         1459         1420
##  LotFrontage  GarageYrBlt GarageFinish   GarageQual   GarageCond   GarageType
##          486          159          159          159          159          157
##     BsmtCond BsmtExposure     BsmtQual BsmtFinType2 BsmtFinType1   MasVnrType
##           82           82           81           80           79           24
##   MasVnrArea     MSZoning    Utilities BsmtFullBath BsmtHalfBath   Functional
##           23            4            2            2            2            2
##  Exterior1st  Exterior2nd   BsmtFinSF1   BsmtFinSF2    BsmtUnfSF  TotalBsmtSF
##            1            1            1            1            1            1
##   Electrical  KitchenQual   GarageCars   GarageArea     SaleType
##            1            1            1            1            1
```

```
cat('There are', length(NAcol), 'columns with missing values')
```

```
## There are 35 columns with missing values
```

We can clearly see that there are 1459 NAs in SalePrice which is matched perfectly with the size of the test set. So, we have to fix NAs in 34 predictor variables.

Missing Value Imputation

We are going to fix the 34 predictors that have missing values in this area. We will go over them one by one, starting with the most common NAs and working our way down until we've repaired them all. If we come across a variable that is part of a group with other variables, we shall treat them as a unit. Pool, Garage, and Basement, for example, each have a number of variations.

We turned character variables into ordinal integers if there is evident ordinality, or into factors if levels are categories without ordinality, in addition to ensuring that NAs are handled. We'll use one-hot encoding to transform these factors to numbers later (using the model.matrix function).

Pool Quality and the PoolArea variable

The PoolQC is the variable with most NAs.

```
kable(unique(all$PoolQC),
      caption= 'Pool Quality Features')
```

Table 3: Pool Quality Features

x
NA
Ex
Fa
Gd

The description is as follows:

PoolQC: Pool quality,Ex:Excellent, Gd:Good,TA: Average/Typical,Fa:Fair, NA:No Pool

So, it is obvious that we need to just assign 'No Pool' to the NAs. Also, the high number of NAs makes sense as normally only a small proportion of houses have a pool.

```
all$PoolQC[is.na(all$PoolQC)] <- 'None'
```

It is also clear that we can label encode this variable as the values are ordinal. As there a multiple variables that use the same quality levels, we are going to create a vector that we can reuse later on.

```
Qualities <- c('None' = 0, 'Fa' = 1, 'Gd' = 2, 'Ex' = 3)
```

Now, we can use the function 'revalue' to do the work for me.

```
all$PoolQC<-as.integer(revalue(all$PoolQC, Qualities))
table(all$PoolQC)
```

```
##
##    0    1    2    3
## 2909    2    4    4
```

However, there is a second variable that relates to Pools. This is the PoolArea variable (in square feet). As you can see below, there are 3 houses without PoolQC. First, I checked if there was a clear relation between the PoolArea and the PoolQC. As I did not see a clear relation (bigger of smaller pools with better PoolQC), I am going to impute PoolQC values based on the Overall Quality of the houses (which is not very high for those 3 houses).

```
ql <- all[all$PoolArea>0 & all$PoolQC==0, c('PoolArea', 'PoolQC', 'OverallQual')]
kable(ql)
```

	PoolArea	PoolQC	OverallQual
2421	368	0	4
2504	444	0	6
2600	561	0	3

Miscellaneous Feature

Within Miscellaneous Feature, there are 2814 NAs. As the values are not ordinal, we will convert MiscFeature into a factor. Values:

```
unique(all$MiscFeature)
```

```
## [1] NA     "Shed" "Gar2" "Othr" "TenC"
```

Elev Elevator

Gar2 2nd Garage (if not described in garage section)

Othr Other

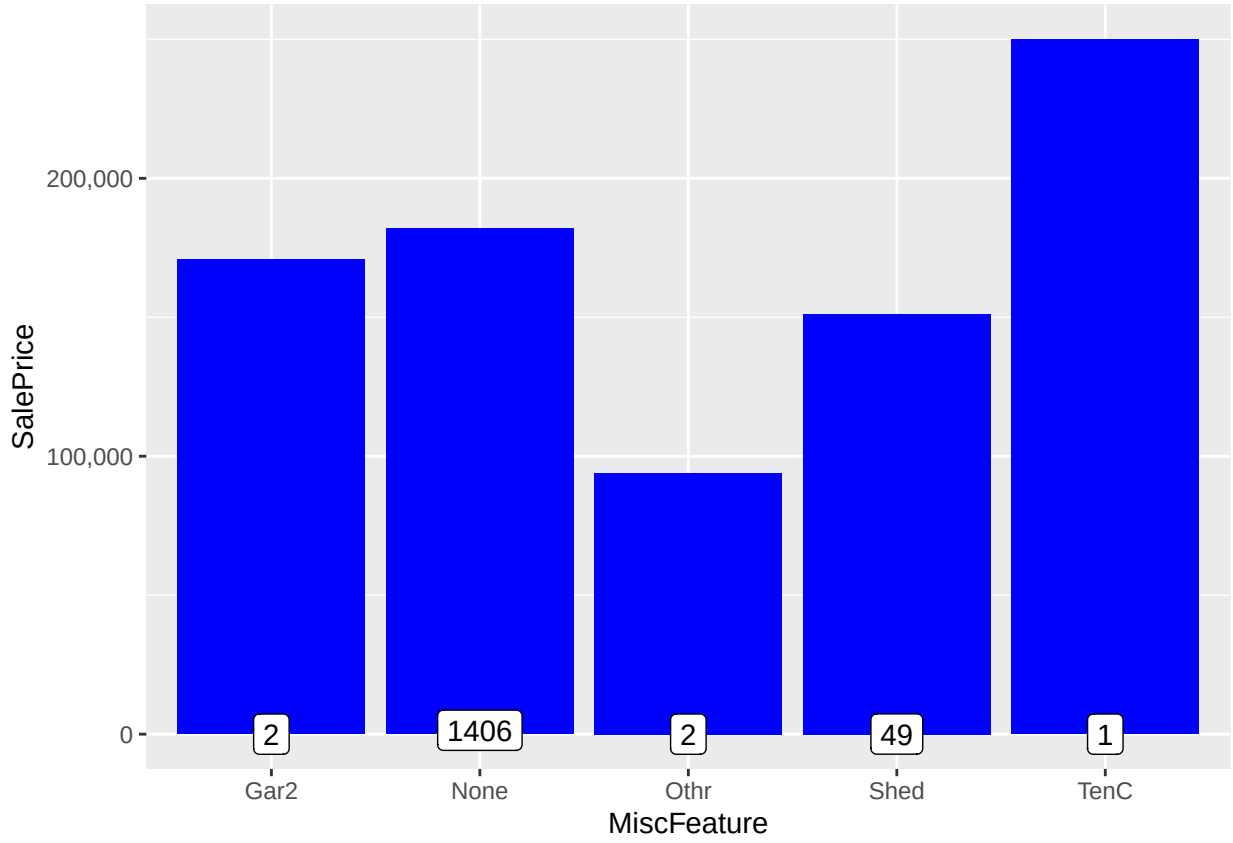
Shed Shed (over 100 SF)

TenC Tennis Court

NA None

```
all$MiscFeature[is.na(all$MiscFeature)] <- 'None'
all$MiscFeature <- as.factor(all$MiscFeature)

ggplot(all[!is.na(all$SalePrice),], aes(x=MiscFeature, y=SalePrice)) +
  geom_bar(stat='summary', fun.y = "median", fill='blue') +
  scale_y_continuous(breaks= seq(0, 800000, by=100000), labels = comma) +
  geom_label(stat = "count", aes(label = ..count.., y = ..count..))
```




```
kable(table(all$MiscFeature),
      caption= 'Table for Miscelenious Features',
      col.names = c('Variable','Frequency'))
```

Table 5: Table for Miscelenious Features

Variable	Frequency
Gar2	5
None	2814
Othr	4
Shed	95
TenC	1

When looking at the frequencies, the variable seems irrelevant to me. Having a shed probably means 'no Garage', which would explain the lower sales price for Shed. Also, while it makes a lot of sense that a house with a Tennis court is expensive, there is only one house with a tennis court in the training set.

Alley Access Type to property

```
unique(all$Alley)
```

```
## [1] NA     "Grvl" "Pave"
```

Within Alley, there are 2721 NAs. As the values are not ordinal, we will convert Alley into a factor.

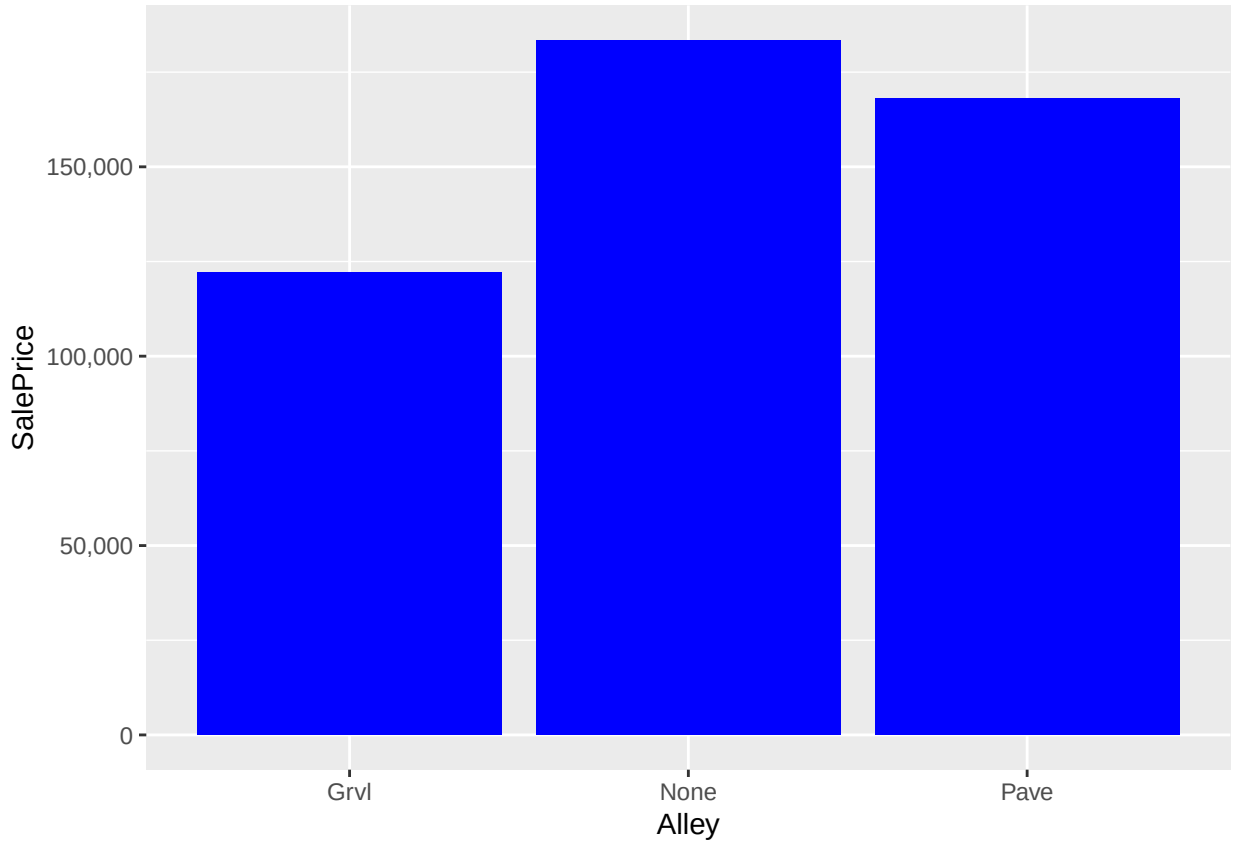
Grvl Gravel

Pave Paved

NA No alley access

```
all$Alley[is.na(all$Alley)] <- 'None'
all$Alley <- as.factor(all$Alley)

ggplot(all[!is.na(all$SalePrice),], aes(x=Alley, y=SalePrice)) +
  geom_bar(stat='summary', fun.y = "median", fill='blue')+
  scale_y_continuous(breaks= seq(0, 200000, by=50000), labels = comma)
```




```
kable(table(all$Alley),
      caption= 'Table for Alley variable',
      col.names = c('Variable','Frequency'))
```

Table 6: Table for Alley variable

Variable	Frequency
Grvl	120
None	2721
Pave	78

Fence Quality

We have seen that, there are 2348 NAs within Fence variable.

```
unique(all$Fence)
```

```
## [1] NA      "MnPrv" "GdWo"  "GdPrv" "MnWw"
```

Meaning

GdPrv Good Privacy

MnPrv Minimum Privacy

GdWo Good Wood

MnWw Minimum Wood/Wire

NA No Fence

```
all$Fence[is.na(all$Fence)] <- 'None'
kable(table(all$Fence),
      caption= 'Table for Fence Quality',
      col.names = c('Type','Frequency'))
```

Table 7: Table for Fence Quality

Type	Frequency
GdPrv	118
GdWo	112
MnPrv	329
MnWw	12
None	2348

```
# Summary Based on median
all[!is.na(all$SalePrice),] %>%
  group_by(Fence) %>%
  summarise(median = median(SalePrice), counts=n())
```

```
## # A tibble: 5 x 3
##   Fence median counts
##   <chr>  <dbl>  <int>
## 1 GdPrv 167500     59
## 2 GdWo  138750     54
## 3 MnPrv 137450    157
## 4 MnWw  130000     11
## 5 None  173000   1179
```

My conclusion is that the values do not seem ordinal (no fence is best). Therefore, I will convert Fence into a factor.

```
all$Fence <- as.factor(all$Fence)
```

Fireplace variables

Fireplace quality

```
all$FireplaceQu[is.na(all$FireplaceQu)] <- 'None'
all$FireplaceQu<-as.integer(revalue(all$FireplaceQu, Qualities))
table(all$FireplaceQu)
```

```
##
##    0    1    2    3
## 1420   74  744   43
```

Values:

Ex Excellent - Exceptional Masonry Fireplace

Gd Good - Masonry Fireplace in main level

TA Average - Prefabricated Fireplace in main living area or Masonry Fireplace in basement

Fa Fair - Prefabricated Fireplace in basement

Po Poor - Ben Franklin Stove

NA No Fireplace

Number of fireplaces

Fireplaces is an integer variable, and there are no missing values.

```
table(all$Fireplaces)
```

```
##
##    0    1    2    3    4
## 1420 1268  219   11    1
```

```
sum(table(all$Fireplaces))
```

```
## [1] 2919
```

###Lot variables

3 variables. One with 1 NA, and 2 complete variables.

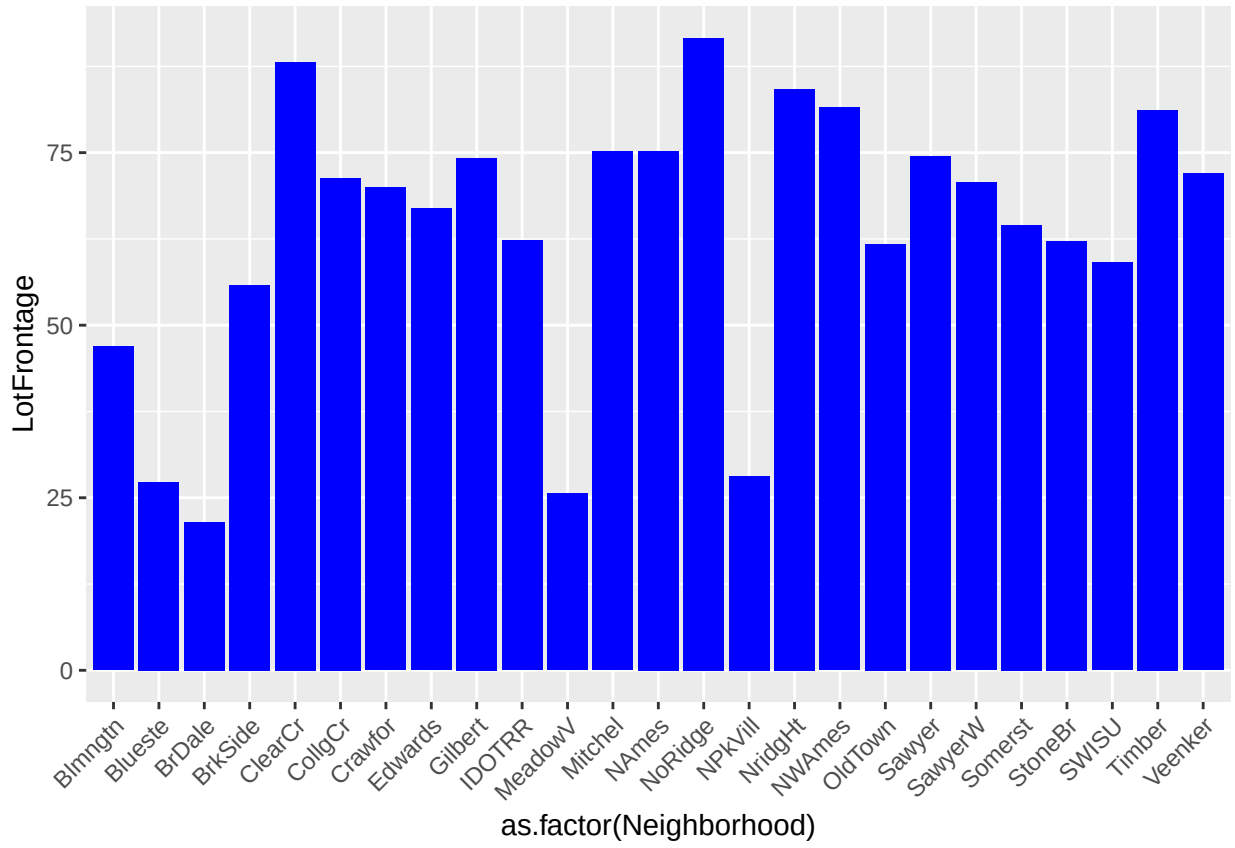
LotFrontage: Linear feet of street connected to property

486 NAs. The most reasonable imputation seems to take the median per neigborhood.

```
ggplot(all[!is.na(all$LotFrontage),], aes(x=as.factor(Neighborhood), y=LotFrontage)) +
  geom_bar(stat='summary', fun.y = "median", fill='blue') +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

```
## Warning: Ignoring unknown parameters: fun.y
```

```
## No summary function supplied, defaulting to `mean_se()`
```



LotShape: General shape of property

No NAs. Values seem ordinal (Regular=best)

Reg Regular

IR1 Slightly irregular

IR2 Moderately Irregular

IR3 Irregular

```
all$LotShape<-as.integer(revalue(all$LotShape, c('IR3'=0, 'IR2'=1, 'IR1'=2, 'Reg'=3)))
table(all$LotShape)
```

```
##
##    0    1    2    3
##   16   76  968 1859
```

```
sum(table(all$LotShape))
```

```
## [1] 2919
```

LotConfig: Lot configuration

No NAs. The values seemed possibly ordinal to me, but the visualization does not show this. Therefore, I will convert the variable into a factor.

Inside Inside lot

Corner Corner lot

CulDSac Cul-de-sac

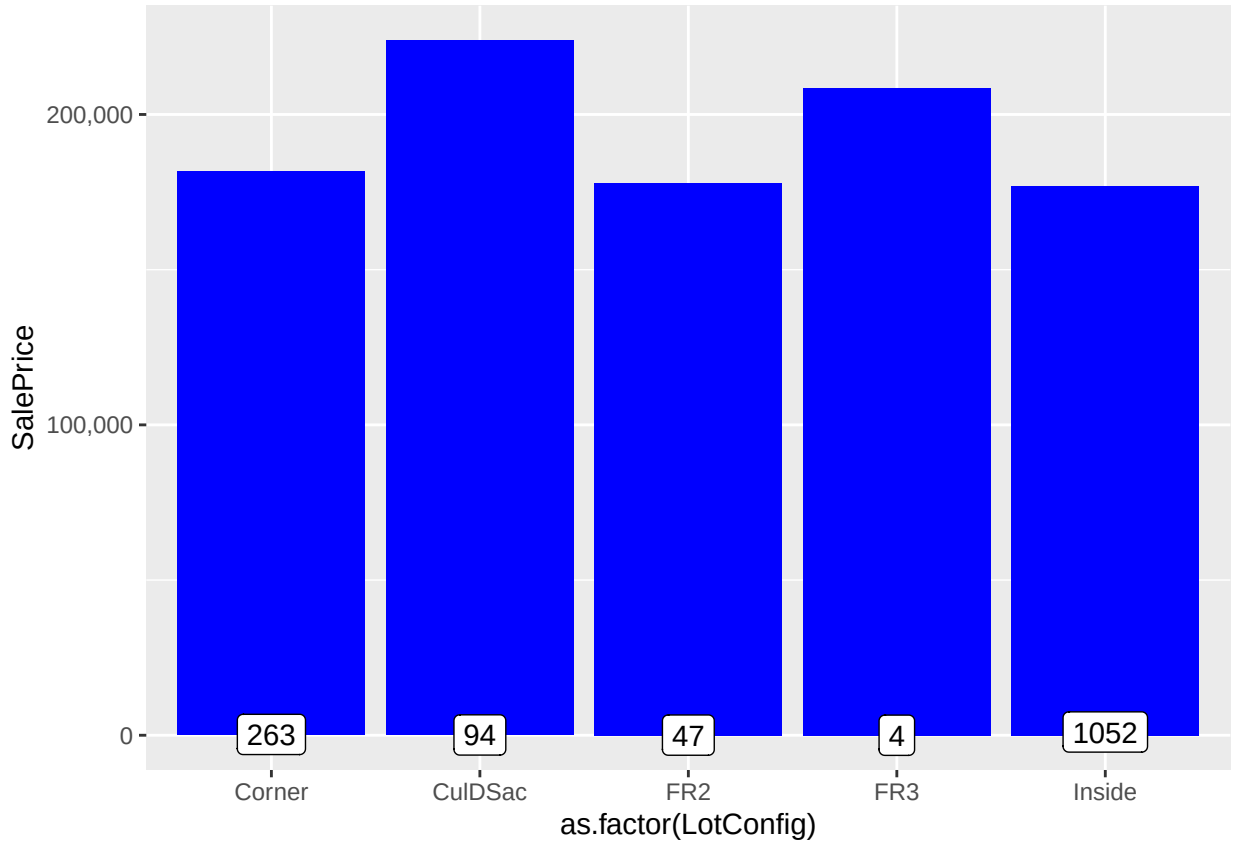
FR2 Frontage on 2 sides of property

FR3 Frontage on 3 sides of property

```
ggplot(all[!is.na(all$SalePrice),], aes(x=as.factor(LotConfig), y=SalePrice)) +
  geom_bar(stat='summary', fun.y = "median", fill='blue')+
  scale_y_continuous(breaks= seq(0, 800000, by=100000), labels = comma) +
  geom_label(stat = "count", aes(label = ..count.., y = ..count..))
```

```
## Warning: Ignoring unknown parameters: fun.y
```

```
## No summary function supplied, defaulting to `mean_se()`
```




```
all$LotConfig <- as.factor(all$LotConfig)
table(all$LotConfig)
```

```
##
##  Corner CulDSac     FR2     FR3  Inside
##     511     176      85      14    2133
```

```
sum(table(all$LotConfig))
```

```
## [1] 2919
```

Garage variables

GarageType GarageYrBlt GarageFinish GarageCars GarageArea GarageQual GarageCond

GarageType: Garage location

The values do not seem ordinal, so I will convert into a factor.

2Types More than one type of garage Attchd Attached to home Basment Basement Garage BuiltIn Built-In (Garage part of house - typically has room above garage) CarPort Car Port Detchd Detached from home NA No Garage

```
all$GarageType[is.na(all$GarageType)] <- 'No Garage'
all$GarageType <- as.factor(all$GarageType)
table(all$GarageType)
```

```
##
##    2Types    Attchd   Basment   BuiltIn   CarPort    Detchd No Garage
##        23      1723        36       186        15       779       157
```

GarageFinish: Interior finish of the garage

The values are ordinal.

Fin Finished

RFn Rough Finished

Unf Unfinished

NA No Garage

```
all$GarageFinish[is.na(all$GarageFinish)] <- 'None'
Finish <- c('None'=0, 'Unf'=1, 'RFn'=2, 'Fin'=3)

all$GarageFinish<-as.integer(revalue(all$GarageFinish, Finish))
table(all$GarageFinish)
```

```
##
##    0    1    2    3
##  159 1230  811  719
```

As from the data we can see that there are 159 houses with no garage, 1230 houses with unfinished garage.

GarageQual: Garage quality

Another variable than can be made ordinal with the Qualities vector.

Ex Excellent

Gd Good

TA Typical/Average

Fa Fair

Po Poor

NA No Garage

```
all$GarageQual[is.na(all$GarageQual)] <- 'None'
Ga.Quality <- c('None' = 0, 'Po' = 1, 'Fa' = 2, 'TA' = 3, 'Gd' = 4, 'Ex' = 5)
all$GarageQual<-as.integer(revalue(all$GarageQual, Ga.Quality))
table(all$GarageQual)
```

```
##
##    0    1    2    3    4    5
##  159    5  124 2604   24    3
```

GarageCond: Garage condition

Garage Condition is another categorical variable which can be converted to integer.

```
all$GarageCond[is.na(all$GarageCond)] <- 'None'
all$GarageCond<-as.integer(revalue(all$GarageCond, Qualities))
```

```
## Warning: NAs introduced by coercion
```

```
table(all$GarageCond)
```

```
##
##   0   1   2   3
## 159  74  15   3
```

MSSubClass

MSSubClass: Identifies the type of dwelling involved in the sale.

20 1-STORY 1946 & NEWER ALL STYLES

30 1-STORY 1945 & OLDER

40 1-STORY W/FINISHED ATTIC ALL AGES

45 1-1/2 STORY - UNFINISHED ALL AGES

50 1-1/2 STORY FINISHED ALL AGES

60 2-STORY 1946 & NEWER

70 2-STORY 1945 & OLDER

75 2-1/2 STORY ALL AGES

80 SPLIT OR MULTI-LEVEL

85 SPLIT FOYER

90 DUPLEX - ALL STYLES AND AGES

120 1-STORY PUD (Planned Unit Development) - 1946 & NEWER

150 1-1/2 STORY PUD - ALL AGES

160 2-STORY PUD - 1946 & NEWER

180 PUD - MULTILEVEL - INCL SPLIT LEV/FOYER

190 2 FAMILY CONVERSION - ALL STYLES AND AGES

These classes are coded as numbers, but really are categories.

```
str(all$MSSubClass)
```

```
##  int [1:2919] 60 20 60 70 60 50 20 60 50 190 ...
```

```
all$MSSubClass <- as.factor(all$MSSubClass)
#revalue for better readability
all$MSSubClass<-revalue(all$MSSubClass, c('20'='1 story 1946+', '30'='1 story 1945-', '40'='1 story unf
kable(table(all$MSSubClass))
```

Var1	Freq
1 story 1946+	1079
1 story 1945-	139
1 story unf attic	6
1,5 story unf	18
1,5 story fin	287
2 story 1946+	575
2 story 1945-	128
2,5 story all ages	23
split/multi level	118
split foyer	48
duplex all style/age	109
1 story PUD 1946+	182
1,5 story PUD all	1
2 story PUD 1946+	128
PUD multilevel	17
2 family conversion	61

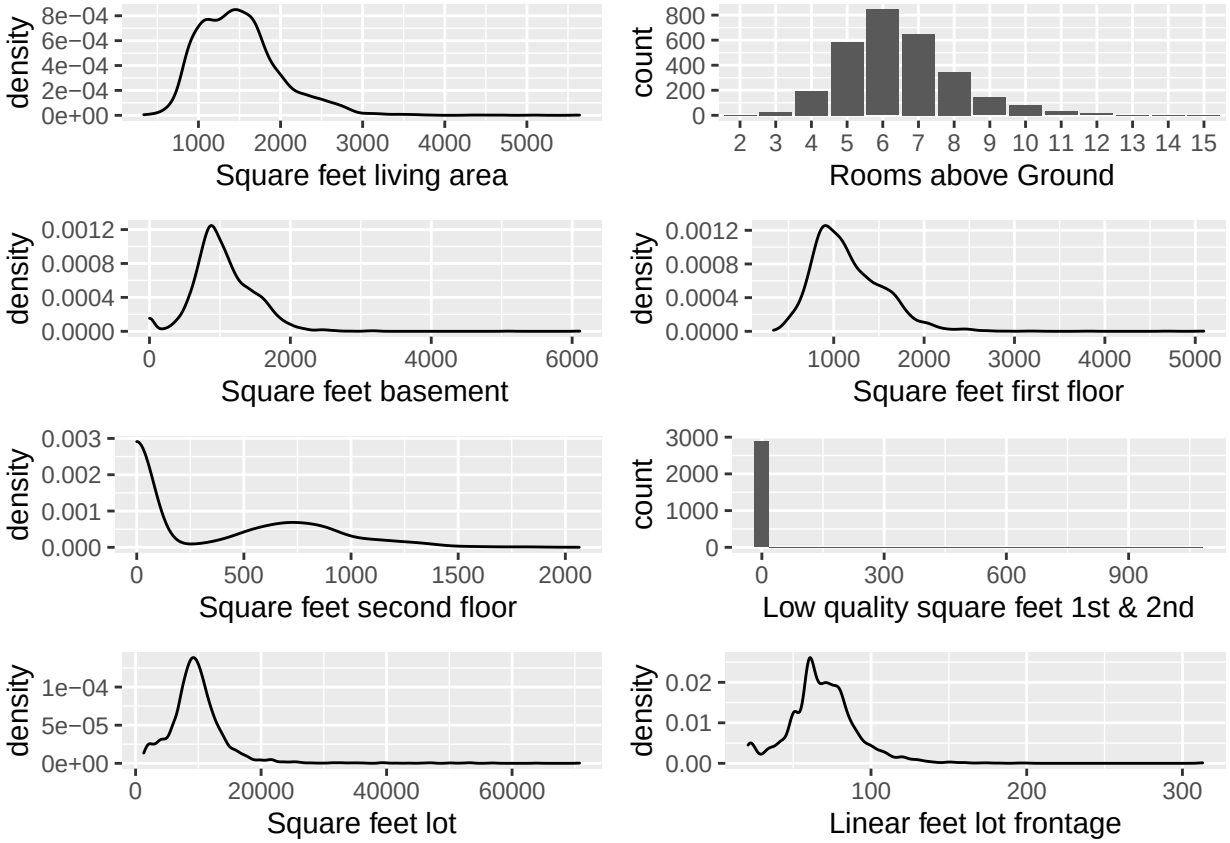
Above Ground Living Area, and other surface related variables (in square feet)

As we have already visualized the relation between the Above Ground Living Area and SalePrice in my initial explorations, we will now just display the distribution itself. As there are more 'square feet' surface measurements in the Top 20, we are taking the opportunity to bundle them in this section. Note: GarageArea is taken care of in the Garage variables section.We are also adding 'Total Rooms Above Ground' (TotRmsAbvGrd) as this variable is highly correlated with the Above Ground Living Area(0.81).

```
sca1 <- ggplot(data= all, aes(x=GrLivArea)) +
 geom_density() + labs(x='Square feet living area')
sca2 <- ggplot(data=all, aes(x=as.factor(TotRmsAbvGrd))) +
 geom_histogram(stat='count') + labs(x='Rooms above Ground')
sca3 <- ggplot(data= all, aes(x=X1stFlrSF)) +
 geom_density() + labs(x='Square feet first floor')
sca4 <- ggplot(data= all, aes(x=X2ndFlrSF)) +
 geom_density() + labs(x='Square feet second floor')
sca5 <- ggplot(data= all, aes(x=TotalBsmtSF)) +
```

```
 geom_density() + labs(x='Square feet basement')
sca6 <- ggplot(data= all[all$LotArea<100000,], aes(x=LotArea)) +
 geom_density() + labs(x='Square feet lot')
sca7 <- ggplot(data= all, aes(x=LotFrontage)) +
 geom_density() + labs(x='Linear feet lot frontage')
sca8 <- ggplot(data= all, aes(x=LowQualFinSF)) +
 geom_histogram() + labs(x='Low quality square feet 1st & 2nd')

layout <- matrix(c(1,2,5,3,4,8,6,7),4,2,byrow=TRUE)
multiplot(sca1, sca2, sca3, sca4, sca5, sca6, sca7, sca8, layout=layout)
```



###The most important categorical variable; Neighborhood

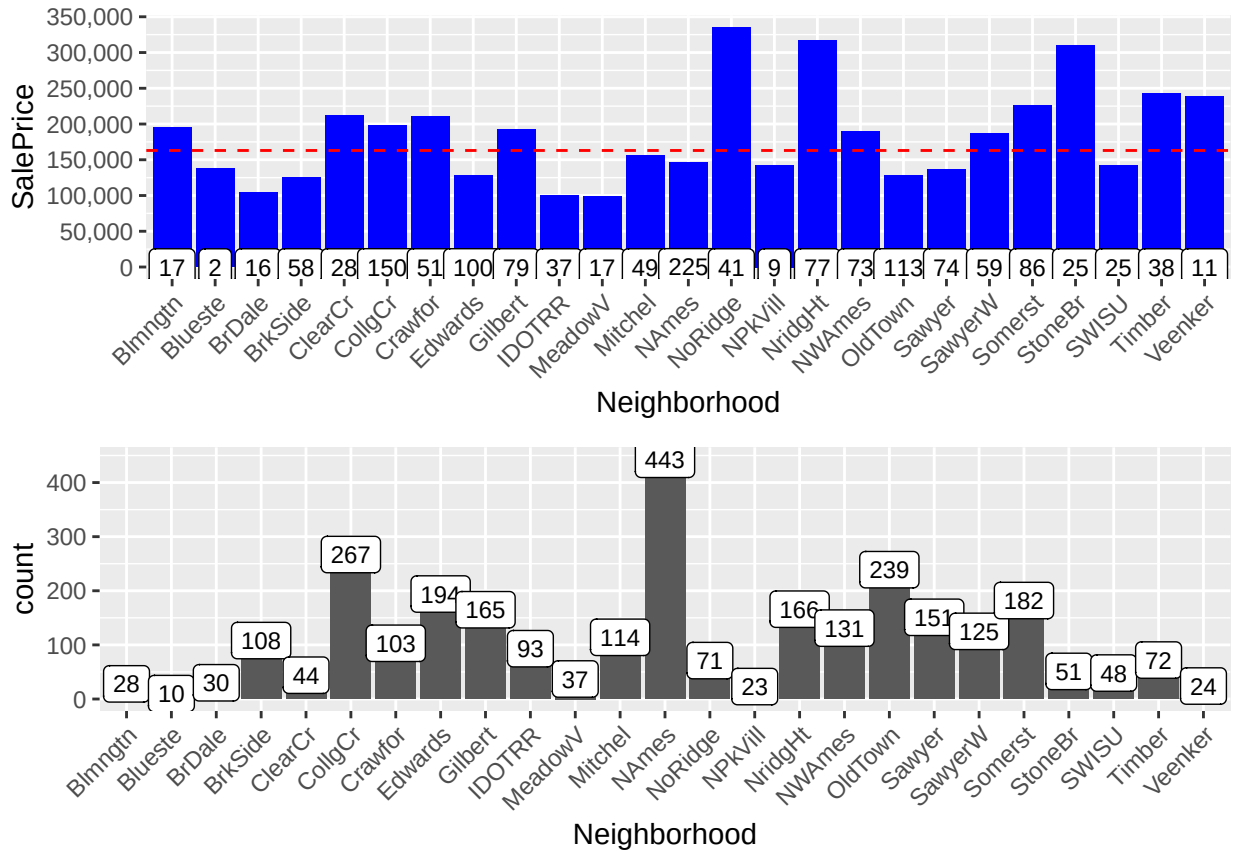
Th first graph shows the median SalePrice by Neighorhood. The frequency (number of houses) of each Neighborhood in the train set is shown in the labels.

The second graph below shows the frequencies across all data.

```
nei1 <- ggplot(all[!is.na(all$SalePrice),], aes(x=Neighborhood, y=SalePrice)) +
 geom_bar(stat='summary', fun.y = "median", fill='blue') +
 theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
 scale_y_continuous(breaks= seq(0, 800000, by=50000), labels = comma) +
 geom_label(stat = "count", aes(label = ..count.., y = ..count..), size=3) +
 geom_hline(yintercept=163000, linetype="dashed", color = "red") #dashed line is median SalePrice
nei2 <- ggplot(data=all, aes(x=Neighborhood)) +
 geom_histogram(stat='count')+
```

```
 geom_label(stat = "count", aes(label = ..count.., y = ..count..), size=3)+
 theme(axis.text.x = element_text(angle = 45, hjust = 1))
grid.arrange(nei1, nei2)
```

```
## No summary function supplied, defaulting to 'mean_se()'
```



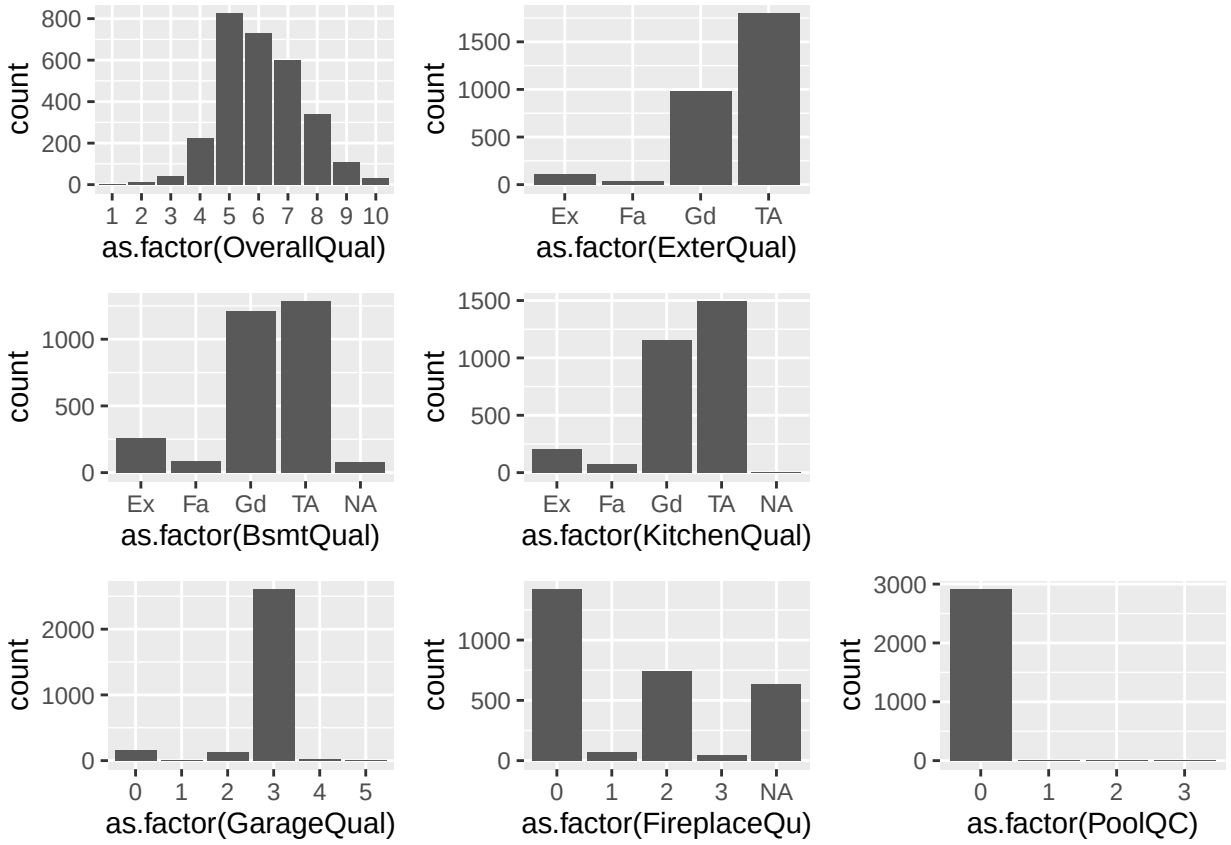
Overall Quality, and other Quality variables

I have already visualized the relation between Overall Quality and SalePrice in my initial explorations, but I want to visualize the frequency distribution as well. As there are more quality measurements, I am taking the opportunity to bundle them in this section.

```
q1 <- ggplot(data=all, aes(x=as.factor(OverallQual))) +
 geom_histogram(stat='count')
q2 <- ggplot(data=all, aes(x=as.factor(ExterQual))) +
 geom_histogram(stat='count')
q3 <- ggplot(data=all, aes(x=as.factor(BsmtQual))) +
 geom_histogram(stat='count')
q4 <- ggplot(data=all, aes(x=as.factor(KitchenQual))) +
 geom_histogram(stat='count')
q5 <- ggplot(data=all, aes(x=as.factor(GarageQual))) +
 geom_histogram(stat='count')
q6 <- ggplot(data=all, aes(x=as.factor(FireplaceQu))) +
 geom_histogram(stat='count')
```

```
q7 <- ggplot(data=all, aes(x=as.factor(PoolQC))) +
 geom_histogram(stat='count')

layout <- matrix(c(1,2,8,3,4,8,5,6,7),3,3,byrow=TRUE)
multiplot(q1, q2, q3, q4, q5, q6, q7, layout=layout)
```



Overall Quality is crucial, yet it is also more detailed than the other characteristics. External quality is important as well, but it has a strong link to overall quality (0.73). Kitchen quality appears to be something to bear in mind, given all residences have a kitchen and there is considerable variation in terms of content. The majority of garages have Q3 quality, which does not appear to make much of a difference. Fireplace Quality is on the list of essential variables and has a good association. The PoolQC is simply lacking (the 13 pools cannot even be seen on this scale).

The second most important categorical variable; MSSubClass

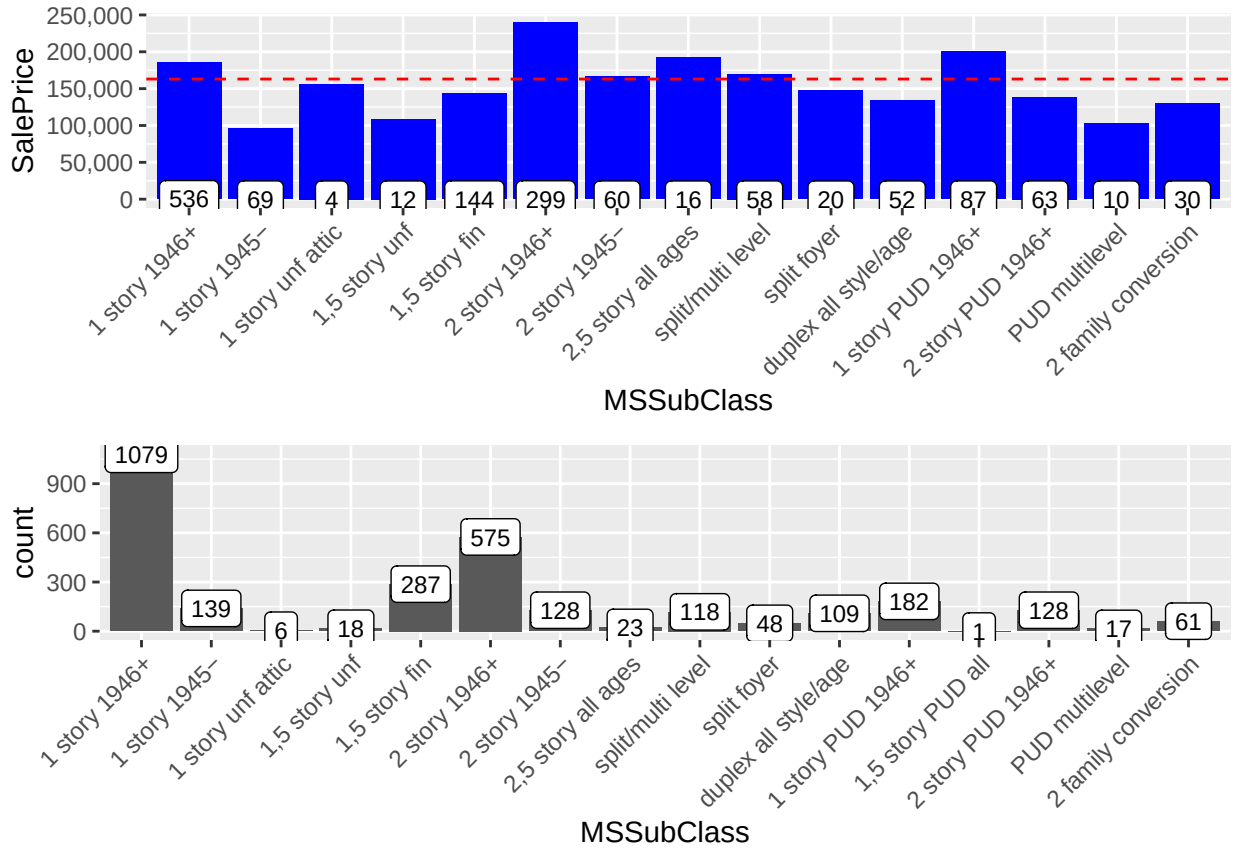
The first visualization shows the median SalePrice by MSSubClass. The frequency (number of houses) of each MSSubClass in the train set is shown in the labels.

The histrogram shows the frequencies across all data. Most houses are relatively new, and have one or two stories.

```
ms1 <- ggplot(all[!is.na(all$SalePrice),], aes(x=MSSubClass, y=SalePrice)) +
 geom_bar(stat='summary', fun.y = "median", fill='blue') +
 theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
 scale_y_continuous(breaks= seq(0, 800000, by=50000), labels = comma) +
```

```
 geom_label(stat = "count", aes(label = ..count.., y = ..count..), size=3) +
 geom_hline(yintercept=163000, linetype="dashed", color = "red") #dashed line is median SalePrice
ms2 <- ggplot(data=all, aes(x=MSSubClass)) +
 geom_histogram(stat='count')+
 geom_label(stat = "count", aes(label = ..count.., y = ..count..), size=3) +
 theme(axis.text.x = element_text(angle = 45, hjust = 1))
grid.arrange(ms1, ms2)
```

```
## No summary function supplied, defaulting to `mean_se()`
```



###Garage variables

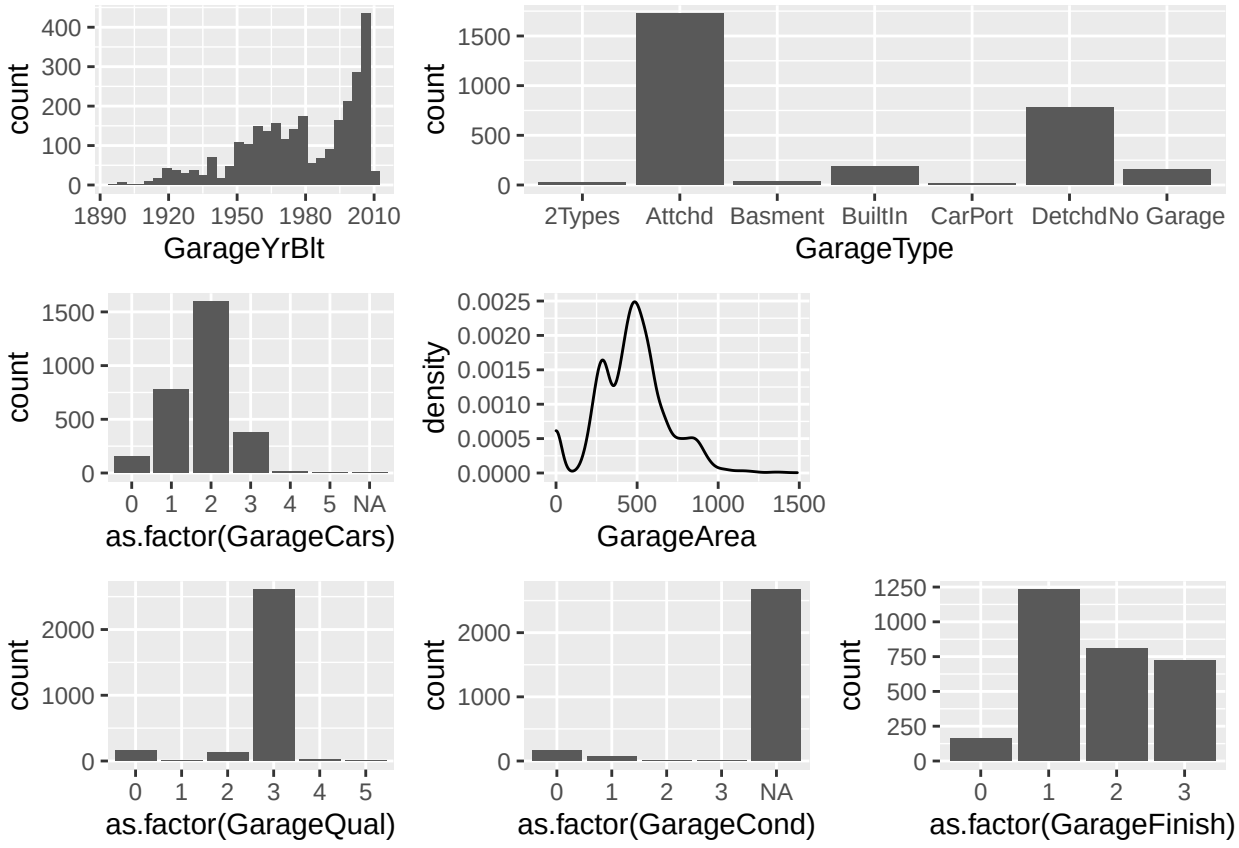
Several Garage variables have a high correlation with SalePrice, and are also in the top-20 list of the quick random forest. However, there is multicolinearity among them and I think that 7 garage variables is too many anyway.We feel that something like 3 variables should be sufficient (possibly GarageCars, GarageType, and a Quality measurement), but before I do any selection I am visualizing all of them in this section.

```
#correct error
all$GarageYrBlt[2593] <- 2007 #this must have been a typo. GarageYrBlt=2207, YearBuilt=2006, YearRemodA
```

```
g1 <- ggplot(data=all[all$GarageCars !=0,], aes(x=GarageYrBlt)) +
 geom_histogram()
g2 <- ggplot(data=all, aes(x=as.factor(GarageCars))) +
 geom_histogram(stat='count')
g3 <- ggplot(data= all, aes(x=GarageArea)) +
```

```
 geom_density()
g4 <- ggplot(data=all, aes(x=as.factor(GarageCond))) +
 geom_histogram(stat='count')
g5 <- ggplot(data=all, aes(x=GarageType)) +
 geom_histogram(stat='count')
g6 <- ggplot(data=all, aes(x=as.factor(GarageQual))) +
 geom_histogram(stat='count')
g7 <- ggplot(data=all, aes(x=as.factor(GarageFinish))) +
 geom_histogram(stat='count')

layout <- matrix(c(1,5,5,2,3,8,6,4,7),3,3,byrow=TRUE)
multiplot(g1, g2, g3, g4, g5, g6, g7, layout=layout)
```



GarageCars and GarageArea are highly correlated. Here, GarageQual and GarageCond also seem highly correlated, and both are dominated by level =3.

Feature engineering

Total number of Bathrooms

There are 4 bathroom variables. Individually, these variables are not very important. However, I assume that I if I add them up into one predictor, this predictor is likely to become a strong one.

"A half-bath, also known as a powder room or guest bath, has only two of the four main bathroom components-typically a toilet and sink." Consequently, I will also count the half bathrooms as half.

```
all$TotBathrooms <- all$FullBath + (all$HalfBath*0.5) + all$BsmtFullBath + (all$BsmtHalfBath*0.5)
```

As you can see in the first graph, there now seems to be a clear correlation (it's 0.63). The frequency distribution of Bathrooms in all data is shown in the second graph.

```
tb1 <- ggplot(data=all[!is.na(all$SalePrice),], aes(x=as.factor(TotBathrooms), y=SalePrice))+
  geom_point(col='blue') + geom_smooth(method = "lm", se=FALSE, color="black", aes(group=1)) +
  scale_y_continuous(breaks= seq(0, 800000, by=100000), labels = comma)
tb2 <- ggplot(data=all, aes(x=as.factor(TotBathrooms))) +
  geom_histogram(stat='count')
grid.arrange(tb1, tb2)
```

```
## `geom_smooth()` using formula 'y ~ x'
```